

Universidad de las Ciencias Informáticas

Facultad # 6



Título: Diseño de la base de datos para el sistema de gestión y transmisión de contenidos audiovisuales.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor(a): Yainery Bolaños Castañeda
Tutor: Ing. Eridniel Suárez Contreras

La Habana, junio 2012

Lo que importa verdaderamente en la vida no son los objetivos que nos marcamos, sino los caminos que seguimos para lograrlo.

Peter Bamm

DECLARACIÓN DE AUTORÍA

Declaración de autoría: Declaro que soy la única autora de este trabajo y autorizo a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yainery Bolaños Castañeda

Estudiante

Ing: Eridniel Suarez Contreras

Tutor

DATOS DE CONTACTO

Tutor: Ing. Eridniel Suárez Contreras

- Profesor graduado de Ing. Informático en el año 2008.
- Ha impartido asignaturas como Sistema de Bases de Datos 1 y Sistemas de Bases de Datos 2, Gestión de Software.
- Posee categoría docente de Instructor.
- Dirección de correo electrónico: esuarez@uci.cu
- Teléfono: 2100

Agradecimientos

A mis padres, por ustedes estoy aquí haciendo realidad mis sueños, un millón de gracias por confiar en mí y estar ahí incondicionalmente, para ustedes van mis logros porque son la razón de mi existir.

A mi hermanita que me ha dado ánimos para seguir adelante, por tí he querido ser siempre un poquito mejor.

A Alberto, por ser mi novio, amigo, hermano, mi brazo derecho. Gracias por darme ánimo y estar ahí siempre y convertírte en parte de mi vida y a su linda familia por acogerme como un miembro más.

A Mirta por sus consejos, por su apoyo, por su preocupación y sobre todo por hacerme sentir como si fuera su nietecita.

A Yeilí por su infinito afecto, cariño y por darme ánimo cuando a veces lo tenía perdido.

A mis primos Edey y Eriecer que son los hermanitos varones que no tengo.

A todos mis tíos que han confiado en mí, por ser tan especiales, sobre todo martica que ha sido otra madre para mí, Sílvio por estar pendiente de mí, de cada prueba, de cada cosita, Félix y fela incomparables por todo su amor, confianza, los quiero mucho y aquí tienen una hijita. A Graciela por ser tan especial por hacerme sentir que soy importante, por tus lindas palabras por todo soy orgullosa de contar con ustedes los quiero mucho son parte de mí.

A mis abuelos Floíran y Enrique por su amor y preocupación en estos añitos de la carrera.

A Luis Alberto por estar siempre ahí en los momentos más difíciles de mi carrera adueñándose de un pedacito de mi corazón y a su mamá por estar al tanto de mí.

Agradecimientos

A Annaliet gracias por aceptarme tal como soy, por sus consejos, preocupaciones, por estar ahí en todo momento, a su linda familia que ha estado también pendiente de mí y a su novio Juan Manuel por estar también tan cerquita de mí brindándome siempre su ayuda.

A Eliana por ser tan especial, por ayudarme, por escucharme, entenderme, ojalá siempre sea así y a su hermana Eliani que ha estado pendiente brindándome todo su apoyo y a su familia en general por ser tan buena.

A Jorge Luis Suarez Falcón porque hay mucho de él en esta tesis, gracias por escucharme y apoyarme cuando más lo necesitaba.

A Amarys y Keily por aconsejarme, apoyarme por estar ahí todos estos añitos de la carrera aun cuando una larga distancia nos separa. Gracias por ser siempre las mismas.

A las buenas amistades que he encontrado en el trascurso de la carrera como a Javier y su madre Ibís que es especial.

A mis compañeros de aula, de apartamento y a todos los que han colaborado para que estos cinco años pasaran como una hermosa experiencia y en específico a Mirita que ha estado pendiente de mí este último añito brindándome todo su apoyo.

A todos los profesores que durante estos años han aportado sus conocimientos, su ayuda, sus consejos para mi formación como son: Abel Días Berenger, Nilverto Sánchez, Yoandry González y en especial a Liesner Acebedo Martínez.

A mi tutor Eridnier, gracias por tu entusiasmo, comprensión por darme ánimo y por ayudarme en la confección de este trabajo.

Al tribunal y oponente gracias por su tiempo, por sus sugerencias y consejos.

Dedicatoria

*A mi abuelita que me vio partir, que lloró y me apoyó, ahora no está presente
para festejar mi logro.*

*A ti va toda mi dedicatoria y aunque no te pueda dar un abrazo y decirte ya
soy ingeniera,*

sé que desde el cielo me estás cuidando y estás orgullosa de mí.

Resumen

Debido al constante desarrollo de la informática y los beneficios asociados a esta, las instituciones han tenido la necesidad de buscar nuevas alternativas para evitar la pérdida y gestión de la información, esta última de una forma más eficiente y segura. El desarrollo de esta ciencia ha provocado que cada una de las esferas de la informática estén inmersas en cambios constantes, el área de la gestión de la información no se encuentra ajena a este proceso. La Universidad de las Ciencias Informáticas (UCI) cuenta con diferentes centros de desarrollo de software, dentro de los cuales se encuentra el centro Geoinformática y Señales Digitales (GEYSED) perteneciente a la facultad 6. El departamento Señales Digitales ubicado en este centro, tiene como propósito la gestión y transmisión de contenidos audiovisuales; sin embargo carece de un mecanismo para llevar satisfactoriamente estos procesos. Esta investigación tiene como objetivo solucionar el problema de la gestión y transmisión de contenidos audiovisuales, esto fue posible mediante el diseño de una base de datos. Como resultado de la investigación se obtuvo una base de datos que responde a las necesidades planteadas inicialmente, se realizó un modelo de datos con alta estructuración y organización de los mismos y se normalizó hasta la tercera forma normal para evitar la redundancia de datos. Para el control de acceso a la información se trazaron estrategias para garantizar la seguridad e integridad de la misma. Finalmente para garantizar la validez de la base de datos se le realizaron pruebas de volumen, carga y estrés.

Palabras claves: base, datos, herramientas, integridad, modelo, validación.

ÍNDICE DE FIGURA

Figura 1 Etapas de diseño de un programa	5
Figura 2 Diagrama de clases persistentes	25
Figura 3 Integridad dominio	35
Figura 4 Integridad referencial	36
Figura 5 Gráfico del resultado de las pruebas.....	43

ÍNDICE DE TABLA

Tabla 1: Archivo_multimedia	27
Tabla 2: Archivo_multimedia_genero	27
Tabla 3: Archivo_multimedia_lista	27
Tabla 4: Archivo_multimedia_lista_node	28
Tabla 5: Canal	28
Tabla 6: Canal_espacio_televisivo	29
Tabla 7: Espacio_dia	29
Tabla 8: Espacio_televisivo.....	29
Tabla 9: Espacio_televisivo_archivo_multimedia	30
Tabla 10: Resultado de la prueba	43
Tabla 11: Resultado de la prueba	43

ÍNDICE DE CONTENIDO

<i>Resumen</i>	VII
<i>Introducción</i>	1
<i>Capítulo 1</i>	5
<i>Fundamentación teórica de la investigación</i>	5
1.1 Introducción	5
1.2 Términos asociados a la investigación.....	5
1.2.1 Base de datos.....	6
1.2.2 Clasificaciones de las bases de datos	6
1.2.3 Sistema de gestión de base de datos	8
1.2.4 Sistemas Gestores de Base de Datos existentes.....	8
1.2.5 Réplicas de bases de datos.....	10
1.2.6 Tipos de entornos de réplica.....	10
1.2.7 Técnicas de replicación	11
1.2.8 Hot Standby y Streaming Replication.....	12
1.3 Análisis de soluciones existentes.....	13
1.3.1 YouTube.....	13
1.3.2 Scribd	13
1.3.3 PRIMICIA 1.6.....	14
1.3.4 Plataforma de Transmisión abierta para Radio y Televisión (PTARTV)	14
1.3.5 Sistema de Captura y Catalogación de Medias (SCCM).....	14
1.3.6 Plataforma VideoWeb.....	14
1.4 Herramientas CASE para el diseño de base de datos. Visual Paradigm.....	15
1.5 Herramienta de desarrollo Pg Admin III para PostgreSQL	15
1.6 Herramienta para prueba de volumen de la base de datos. EMS Data Generator 2005 para PostgreSQL.....	16

ÍNDICE DE CONTENIDOS

1.7	La herramienta JMeter para la validación del diseño de la base de datos.....	17
1.8	Conclusiones parciales	17
<i>Capítulo 2</i>		19
<i>Descripción y análisis de la solución propuesta</i>		19
2.1	Introducción.....	19
2.2	Descripción de la arquitectura	19
2.3	Requisitos funcionales y no funcionales	21
2.4	Diseño de la base de datos	23
2.4.1	Modelo lógico y físico de la base de datos	24
2.4.2	Descripción de las tablas	26
2.5	Optimización de la base de datos.....	30
2.6	Conclusiones parciales.....	32
<i>Capítulo 3</i>		33
3.1	Introducción	33
3.2	Integridad de los datos.....	33
3.3	Integridad de entidad	34
3.4	Integridad de dominio.....	34
3.5	Integridad referencial	35
3.6	Integridad semántica.....	36
3.7	Normalización de la base de datos	37
3.8	Análisis de la redundancia de la información	38
3.9	Análisis de la seguridad y recuperación	38
3.10	Análisis de la replicación de datos.	40
3.11	Validación de la base de datos	41
3.12	Trazabilidad de las acciones	44

ÍNDICE DE CONTENIDOS

3.13 Conclusiones parciales	45
<i>Conclusiones Generales</i>	46
<i>Recomendaciones</i>	47
<i>Bibliografía</i>	48
<i>Referencias Bibliográficas</i>	51
<i>Glosario de términos</i>	54

Introducción

El desarrollo y avance de la comunicación está vigente en la actualidad; los procesos de transmisión y recepción de contenidos, información y mensajes, siguen constituyendo una necesidad en todos los tiempos. Las Tecnologías de Información y Comunicación (TIC) han evolucionado, convirtiéndose en un eslabón fundamental para la gestión de la información y los recursos. En sus inicios los datos se almacenaban manualmente, este era un trabajo engorroso y lento. A mediados de los años 80 del siglo XX con la aparición de los primeros ordenadores personales y las organizaciones con archivos, bibliotecas y centros de documentación, comenzó un proceso de cambio generalizado, excitando la transición del papel al formato electrónico como una forma de producir, almacenar y recuperar la información, lo que provocó más tarde que apareciera el término de bases de datos electrónicas.

El surgimiento de las bases de datos se remonta en el año 1860 por Herman Hollerit, quien inventó la primera “máquina automática perforadora de tarjetas” con el funcionamiento básico de una computadora para hacer el censo poblacional de Estados Unidos. Más tarde, en 1950 surgen las cintas magnéticas, las cuales fueron el elemento fundamental para automatizar la información de las nóminas, a través de la lectura y copia de datos entre cintas, además se podían transferir a las tarjetas perforadas, simulando un sistema de salvos. A partir del año 1960 se empezaron a emplear los discos duros de computadoras, como soporte para la gestión de la información. (Baixauli, 2005). En 1970 Edgar Frank Codd planteó la idea de un modelo relacional, exponiendo una forma de organizar las bases de datos mediante un patrón matemático lógico, a partir de dicho patrón, surge el lenguaje SQL (Lenguaje Estructurado de Consultas) que se utiliza actualmente para generar reportes, consultar, actualizar y administrar la información, esto permitió trabajar con datos más complejos. (Anzaldo, 2005)

El desarrollo de las aplicaciones multimedia ha dado un giro al concepto base de datos, que sólo integraba elementos de información textual y numérica dando lugar a tipos de objetos como son: gráficos, sonoros y audiovisuales que comparten el mismo entorno que los datos textuales y numéricos. Las bases de datos multimedia tienen la peculiaridad de que los datos pueden tener variabilidad espacial y temporal.

La UCI cuenta con diferentes centros de desarrollo de software dentro de los cuales se encuentra el centro GEYSED perteneciente a la facultad 6. El departamento Señales Digitales ubicado en este centro tiene como perspectivas el desarrollo de sistemas informáticos orientados a la transmisión automatizada de contenidos audiovisuales, contando con productos que realizan sus funciones a partir de un contenedor de información capaz de manipular los datos que utilizan. Ejemplos de estos productos son:

- Plataforma de Televisión Informativa (PRIMICIA versión 1.6): transmite noticias para los canales de televisión de la entidad o del lugar donde se despliegue. Permite la administración del canal, la gestión de la información y los recursos de multimedia y materiales que son publicados.
- VideoWeb: sistema para la transmisión de contenido multimedia sobre la red de datos. Permite la administración, la gestión de los contenidos audiovisuales a publicar y de secciones de información, habilitación e inhabilitación de contenidos y las búsquedas.
- Sistema de Captura y Catalogación de Medias (SCCM): es una herramienta automatizada de almacenamiento de archivos audiovisuales. Considerado una solución satisfactoria para la captura de señales de video y audio según planificaciones definidas, almacenamiento y gestión de los ficheros de video capturados en un archivo y la catalogación de los materiales audiovisuales almacenados.
- Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV): es una herramienta que integra los tipos más comunes de transmisiones que existen, dígase audio, video, teletexto, y prensa escrita. Provee de una solución tecnológica integral que permita automatizar la transmisión, administración y gestión de los canales de radio y de televisión que pudieran existir en una entidad.

Este departamento, no posee un contenedor de información que le permita a sus productos, la organización, control y manipulación de la información que utilizan, de forma centralizada, situación que lleva a plantearse el siguiente **problema a resolver**: ¿Cómo contribuir a la gestión y manipulación de los datos utilizados en un sistema de gestión y transmisión de contenidos audiovisuales? Para darle solución a lo antes planteado, se propone como **objetivo general**, diseñar una base de datos que permita manipular los datos de un sistema de gestión y transmisión de contenidos audiovisuales.

Surgiendo como **objeto de estudio** el diseño e implementación de la base de datos relacional enmarcado en el **campo de acción** correspondiente a la gestión de la información que manipula un sistema de gestión y transmisión de contenidos audiovisuales.

Se plantea como idea a defender que con el diseño de la base de datos, se permitirá la gestión de la información de forma segura, evitando la redundancia y garantizando la integridad.

Para darle cumplimiento al objetivo planteado, se trazan las siguientes tareas de la investigación:

- Caracterizar los procesos de gestión y manipulación de datos en la plataforma de transmisión de contenidos multimedia, así como las herramientas para el diseño e implementación de la base de datos.

- Elaborar el diseño lógico, así como el esquema relacional y sus puntos de integración de la base de datos para la plataforma de transmisión de contenidos multimedia.
- Implementar una política de respuesta a fallos del sistema para evitar la pérdida de información.
- Definir la distribución física del esquema de base de datos en los servidores (nodos activos de la red).
- Elaborar el diseño físico de la base de datos, así como optimizar el rendimiento de la misma.
- Definir e implementar las políticas de recuperación y salvadas automáticas de la base de datos.
- Validar el funcionamiento de la base de datos a partir de pruebas durante el funcionamiento.

Para dar cumplimiento a las tareas de la investigación, se tuvieron en cuenta un grupo de métodos, tanto teóricos como empíricos.

Métodos teóricos

Analítico – sintético: método utilizado en el análisis de la bibliografía consultada y la realización de síntesis sobre la información relevante.

Análisis Histórico – Lógico: método utilizado para describir las tendencias actuales de creación de bases de datos, así como el estudio y análisis de las principales herramientas para el manejo de datos reproduciendo lo esencial.

Modelación: método empleado para describir en diagramas, mediante los componentes que los integran, la dinámica de los procesos e interacción entre objetos a partir de sus relaciones.

Métodos empíricos.

Entrevistas: método empleado con el objetivo de obtener información de especialistas con conocimientos, sobre la manipulación de información, así como el trabajo con bases de datos.

El trabajo de diploma está compuesto de 3 capítulos:

Capítulo 1. Fundamentación teórica: en este capítulo se recogen los principales conceptos y definiciones que deben conocerse antes de realizar el diseño y la implementación de la base de datos. Se describen las herramientas a utilizar para el diseño y se analizan las soluciones existentes tanto en el ámbito internacional como en el nacional.

Capítulo 2. Descripción y análisis de la solución propuesta: en este capítulo se hará la especificación de los requisitos a tener en cuenta durante el desarrollo de la propuesta de solución, se realizará el análisis de la aplicación, la descripción de la arquitectura y el diseño de la base de datos. Se optimizará la base de datos basándose en los componentes de software y hardware.

Capítulo 3. Validación de la propuesta: en este capítulo se presenta la implementación de la base de datos, las políticas de salva y recuperación ante fallos y la validación funcional de la misma, aplicando pruebas de volumen, carga y estrés.

Capítulo 1

Fundamentación teórica de la investigación

1.1 Introducción

En este capítulo se analizan temas relacionados con las bases de datos, el lector tendrá a su disposición definiciones, conceptos, características de sistemas gestores, interfaces y tipologías. Se expone un conjunto de soluciones existentes tanto en Cuba como en el resto del mundo sobre el diseño de las bases de datos, lo cual permitirá detallar las principales tendencias y tecnologías sobre las que se apoyará la propuesta final.

1.2 Términos asociados a la investigación

El desarrollo de un proyecto transita por un grupo de etapas en su ciclo de vida como se muestra en la figura #1. Primeramente se encuentran los datos, que es el eslabón fundamental a la hora de realizar cualquier tarea. La segunda es el proceso, que incluye un conjunto de operaciones o tareas que se realizan en un orden específico y definido. La última etapa es referente a la información, que es el resultado obtenido al ser procesados los datos; el objetivo es adquirir información segura y precisa en el menor tiempo posible. Los datos pueden estar en movimiento o en reposo, estos últimos hay que almacenarlos en archivos, ficheros o bases de datos. (Rozic, 2004)

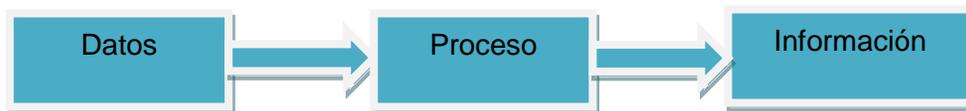


Figura 1: Etapas de diseño de un programa

Inicialmente la información se almacenaba en archivos, estos estaban compuestos por registros y cada registro por un conjunto de campos. Los archivos se organizaban de tres formas diferentes: (Rozic, 2004)

- Archivos secuenciales (se hace uno tras otro).
- Archivos indexados (se anexa el archivo al índice).
- Archivos relativos o directos (se determina su posición física).

Los archivos tienen algunos inconvenientes como son:

- Redundancia de datos.
- Problema de consistencia e integridad de los datos.

Capítulo 1. Fundamentación teórica de la investigación

- Dificultad para modificar la estructura lógica de los datos.

Dadas estas dificultades surge la necesidad de una nueva forma de almacenamiento de datos, que además de ser robusta, permite su manipulación y modificación según las necesidades del usuario, es por eso que aparece el término “base de datos”.

1.2.1 Base de datos

Una base de datos es un sistema formado por un conjunto de datos almacenados en un soporte legible, organizados internamente por registros y campos que permiten el acceso directo, y por programas que manipulen ese conjunto de datos. Permiten recuperar cualquier variedad de información, ya sea referencias, documentos textuales, imágenes o datos estadísticos. (Anzaldo, 2005). Una base de datos se le puede denominar como el lugar donde se almacenan los datos en reposo y donde se pueden acceder desde diferentes aplicaciones sistemas o programas. (Rozic, 2004). También el término de base de datos se puede manejar como un conjunto de información o de datos relacionados que se encuentran agrupados o estructurados. (Camallea, 2004).

Otra forma de identificar una base de datos es como un conjunto exhaustivo de datos estructurados, fiables y homogéneos, organizados independientemente de su utilización e implementación en una computadora, accesibles en tiempo real, que pueden compartir varios usuarios con necesidades de información diferentes y no predecibles en el tiempo, es decir, una base de datos es una colección de datos estructurados independientemente de la aplicación y del soporte de almacenamiento que los contiene, que presente la menor redundancia posible y que además puedan ser compartidos con varios usuarios. (Camallea, 2004).

Por tanto una base de datos engloba a la información concerniente, disponibles para los usuarios, con la finalidad de eliminar la redundancia o al menos minimizarla. Se puede entender como el lugar donde se almacenan una serie de datos organizados, relacionados entre sí y donde estos serán accesibles desde cualquier aplicación.

1.2.2 Clasificaciones de las bases de datos

Existen diferentes bases de datos, estas se pueden clasificar según su utilidad, necesidad o de acuerdo al contexto que se esté manejando. (Barajas, 2009)

Según la variabilidad de los datos almacenados pueden clasificarse en:

Capítulo 1. Fundamentación teórica de la investigación

- Dinámica: la información contenida en la base de datos puede ser modificada con el tiempo, lo cual permite operaciones como actualizar, eliminar, adicionar datos y realizar consultas.
- Estática: la información contenida en la base de datos es solo de lectura, después se puede utilizar para realizar proyecciones y tomar decisiones. (Barajas, 2009)

Según su contenido en:

- Bibliográficas: contiene información ya sea sobre el autor, editorial, título, edición o fecha de determinada publicación de un libro.
- De directorios: son guías electrónicas que ofrecen datos personales de usuarios.
- De texto completo: almacenan la información de cualquier colección de un autor.

Según su función se pueden clasificar de acuerdo a su modelo de administración de datos. Los modelos permiten tener una aproximación del diseño de las bases de datos, describen los elementos, sus relaciones y las características fundamentales de una situación determinada. Se pueden clasificar como:

Modelos lógicos basados en objetos: son utilizados para describir datos de la misma forma que son captados en el mundo real, son flexibles y permiten especificar restricciones de datos explícitamente. Existen numerosos modelos de este tipo pero el más conocido y utilizado por su sencillez es el Entidad-Relación. Representa la realidad a través de entidades, busca las entidades básicas del modelo y luego la relación que existe entre ellas.

Modelos lógicos basados en registros: son manejados para describir datos en los niveles conceptual y físico. Se emplea para especificar la estructura lógica global de la base de datos y facilitar una descripción a nivel más alto de la implementación; a través de registros e instancias. Los modelos más aceptados en esta clasificación son:

- Modelo Relacional: es un conjunto de relaciones donde todos los datos visibles están representados como tablas de valores, en las cuales las filas corresponden a los registros y los atributos o características equivalen a las columnas de cada registro.
- Modelo de Red: son registros lógicos que se representan mediante colecciones de registros o entidades y que se relacionan entre sí por medio de flechas.
- Modelo Jerárquico: los datos y las relaciones son similares al modelo de red, con la diferencia que en lugar de estar organizados por gráficas arbitrarias, lo hacen por conjuntos de árboles.

Capítulo 1. Fundamentación teórica de la investigación

- **Modelos físicos:** se debe conocer el manejador de bases de datos y se debe saber cuál es el sistema de gestión de bases de datos a usar, pues este modelo se tiene que adaptar a él. (Barajas, 2009)

Según los términos abordados, se puede decir, que sería óptima una base de datos de tipo dinámica, por las grandes facilidades que brinda como modificar y actualizar los datos. Se define que el modelo más factible es el relacional, que está dentro de los lógicos basados en registros, pues favorece a la normalización y no tiene en cuenta el orden de los datos.

1.2.3 Sistema de gestión de base de datos

Un sistema gestor de base de datos (SGBD), es un tipo de software específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan; lo que es equivalente a una agrupación de programas que sirven para definirla, construirla y manipularla, permitiendo almacenar y posteriormente acceder a los datos de forma rápida y estructurada. (Anzaldo, 2005)

Los SGBD deben cumplir con un grupo de objetivos, ellos son:

- **Abstracción de la información:** se definen varios niveles de abstracción es transparente al usuario, ahorrando detalles de almacenamiento físico de los datos.
- **Independencia:** da la posibilidad de modificar los esquemas físico o lógico de una base de datos sin necesidad de realizar cambios en las aplicaciones que se sirvan de ellas.
- **Consistencia:** permite que en los casos que no se haya podido eliminar la redundancia, se vigile que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad:** permite otorgar diversas categorías de permisos para los usuarios y grupos.
- **Manejo de transacciones:** facilitan mecanismos para programar las modificaciones de los datos de forma simple.
- **Tiempo de respuesta:** permite establecer el tiempo que demora en facilitarse la información solicitada y en almacenarse los cambios realizados.

1.2.4 Sistemas Gestores de Base de Datos existentes

Existen numerosos SGBD, los de mayor relevancia son:

Oracle: es un SGBD relacional desarrollado por Oracle Corporation, utilizado para trabajar con grandes volúmenes de datos, suele ser manejado en intranet y sistemas de gran alcance. Es usado por biblioteca

Capítulo 1. Fundamentación teórica de la investigación

multimedia para el manejo de información personal, capaz de ejecutarse en más de 80 arquitecturas de hardware y software distintos sin realizar modificaciones, soporta la mayoría de los idiomas computacionales, asimila datos alfanuméricos, textos sin estructura, imágenes, audio, video y está disponible en múltiples plataformas como Windows, Linux y en las versiones de Unix. Es un sistema de gran estabilidad, seguridad, soporte de transacciones y escalabilidad. Dentro de sus desventajas está el costo elevado por soporte técnico y que constituye un software propietario, su licencia es una de las más caras en el mundo. (Arévalo, 2010)

MySql: es patentado bajo la licencia GPL para cualquier uso compatible de ella, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Su diseño multihilo, le permite soportar una gran carga de forma eficiente. Es de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Cuenta con abundantes librerías, que permiten su uso a través de múltiples lenguajes de programación, es fácil de instalar y configurar. También se distribuye una versión comercial, que solo se diferencia en el soporte técnico que se ofrece. Posee las siguientes desventajas: presenta cierta limitación en cuanto al trabajo con disparadores, los tipos de datos son reducidos, es fácil de vulnerar sin protección adecuada y carece de capacidades de integridad relacional. (Arévalo, 2010)

Microsoft SQL Server: basa su lenguaje en Transact-SQL. Utilizado para manejar grandes volúmenes de información de manera simultánea. Provee a la base de datos de escalabilidad, estabilidad y seguridad, incluye un potente entorno gráfico de administración y facilita administrar información de otros servidores de datos. Tiene algunas desventajas como son: no emplea una administración dinámica de memoria, no maneja compresión de datos, por lo que las bases de datos pueden llegar a ocupar mucho espacio en disco, requiere de un sistema operativo Microsoft Windows, por lo que no puede instalarse, por ejemplo, en servidores Linux. (Centro de desarrollo de SQL Server, 2009).

PostgreSQL: es un SGBD relacional de software libre. Contiene características de la orientación a objetos, como puede ser tipos de datos, herencia, funciones, restricciones, disparadores, reglas e integridad transaccional. Está disponible bajo una Licencia BSD, puede tener uso tanto comercial como no comercial, lleva más de 15 años desarrollándose y su arquitectura goza de una excelente reputación por su fiabilidad e integridad de los datos. Presenta valiosas características como permite la replicación asincrónica, transacciones anidadas, copias de seguridad en caliente, un sofisticado planificador/optimizador de consultas, un registro nativo de escrituras anticipadas para ser tolerantes a

Capítulo 1. Fundamentación teórica de la investigación

fallos de hardware y presenta un control concurrente multiversión (MVCC). Actualmente es el más utilizado por la comunidad de software libre después de MySQL, aunque presenta una estructura superior a este y propone más funcionalidades, es escalable tanto en la cantidad bruta de datos que puede manejar como en el número de usuarios concurrentes que puede atender. (Thomas Lockhart).

Después del análisis realizado, se decidió utilizar PostgreSQL como SGBD para desarrollar la base de datos del sistema de gestión y transmisión de contenidos audiovisuales, significándose su potencia y libertad de uso, esto contribuye a la soberanía tecnológica del país. Presenta además dos elementos de importancia incluidos desde la versión 9.0 de PostgreSQL, estos son Hot Standby (HS) y Streaming Replication (SR) para permitir el trabajo con réplicas.

1.2.5 Réplicas de bases de datos

Partiendo de la conceptualización de red, como varios ordenadores y dispositivos enlazados para tener facilidades de comunicación, se denomina réplica a un sistema de administración de bases de datos distribuida, este es el proceso de copiar la base de datos en otros lugares de la red lo que posibilita que estén sincronizadas. (Ostorga, 2010)

La réplica de datos debe caracterizarse por: (Ostorga, 2010)

- Efectividad: depende de cómo los datos son distribuidos y almacenados, ya que si la réplica se realiza positivamente entonces la disponibilidad de datos para ejecutar procesos paralelos será mayor.
- Disponibilidad: es el tiempo en el que un servicio puede ser accedido. Debe existir un servidor adicional que posea alguna técnica de replicación que lo pueda suplantar en caso de ser necesario.
- Tolerancia a fallos: es fundamental para garantizar el comportamiento correcto, donde pueden existir un número finito de fallos y sus tipos.
- Coordinación: se realizan las invocaciones de los servicios a los objetos, que al final de la transacción debe realizarse tal y como fue solicitada, para lo cual debe utilizar algún tipo de ordenamiento.

1.2.6 Tipos de entornos de réplica

Las réplicas se hacen de acuerdo a las necesidades propias. Existen dos maneras diferentes para su aplicación: (Iborra, 2004)

Capítulo 1. Fundamentación teórica de la investigación

- Maestro-Esclavo (*master-slave*): es solo de lectura, facilita a un solo maestro recibir consultas de lectura/escritura mientras los esclavos solo son los encargados de aceptar consultas de lectura.
- Maestro-Maestro (*multi-master*): llamado par-a-par o la réplica de camino de n, da la posibilidad de múltiples sitios, actuando como pares iguales. Cada sitio es considerado como un sitio maestro y se comunican con otros, de esta forma se sincronizan los cambios entre los servidores. Es fundamental para mantener sitios recuperables ante posibles desastres o caídas, así como para suministrar sistemas con alta disponibilidad y balancear la carga de consultas a través de las distintas ubicaciones.

1.2.7 Técnicas de replicación

Existen dos modelos de distribución de datos esencialmente utilizados en la replicación:

- Asincrónico: almacena y reenvía, se encarga de capturar cualquier cambio local, los almacena en una cola a intervalos regulares, propaga y aplica estos cambios en sitios remotos.
- Sincrónico: replica en tiempo real, aplica cualquier cambio o establece cualquier procedimiento reproducido en todos los sitios que participan en el ambiente de réplica como parte de una sola transacción por lo que asegura la consistencia de los datos.

Estos modelos tienen asociados técnicas de replicación: (Vega, 2007)

Sincrónica:

- Confirmación en dos fases: facilita la sincronización de datos distribuidos.

Asincrónica:

- Descarga y Recarga (*Dump and Reload*): encargado de copiar la salva para un dispositivo de almacenamiento y después distribuirla por los demás servidores. Presenta inconvenientes como que se realiza de forma manual y se consultan datos desactualizados.
- Instantánea (*Snapshot*): es como captar una imagen de los valores publicados en la base de datos en un instante determinado. Permite la consistencia entre el publicador que es el servidor que hace los datos disponibles para la réplica hacia otros servidores y el suscriptor que son los servidores que almacenan las reproducciones y reciben actualizaciones. Proporciona inconvenientes como poca autonomía en el sitio, se limita su utilidad a un ambiente sin conexión, las tablas esclavas son solo de lecturas y mientras más alta sea la frecuencia de reajuste es más limitada la solución.

Capítulo 1. Fundamentación teórica de la investigación

- Transaccional: se encarga de supervisar los cambios al servidor que publica en el nivel de la transacción. Los realiza en tiempo real, con solo segundos de diferencias, estos se deben efectuar en el sitio en que se publica para evitar conflictos y garantizar la consistencia transaccional.

Una vez caracterizado el proceso de réplica de base de datos, se decide aplicar la técnica asincrónica, permitiendo dicha técnica, la copia y distribución de una base de datos y sus objetos de forma idéntica en múltiples ordenadores, asegurando que los datos estén disponibles en varias localizaciones.

1.2.8 Hot Standby y Streaming Replication.

PostgreSQL a partir de su versión 9.0 tiene incluida dos características que son fundamentales para el trabajo con réplicas, Hot Standby y Streaming Replication, estas tienen la peculiaridad de implementar en el núcleo de PostgreSQL lo necesario para instalar un sistema de replicación asincrónica maestro-esclavo en el que los nodos esclavos se pueden utilizar para realizar consultas de solo lectura, brindando la posibilidad de añadir redundancia a la base de datos y permite aliviar de trabajo al servidor principal en lo referente a consultas de solo lectura.

Streaming Replication transfiere de forma asincrónica registros WAL¹ sobre la marcha (*record-based log shipping*) entre un servidor maestro y uno o varios esclavos. Esta técnica se realiza mediante un proceso denominado receptor WAL en el servidor esclavo, se conecta mediante una conexión TCP/IP al servidor maestro. En el servidor maestro existe otro proceso denominado remitente WAL que es el encargado de mandar los registros WAL sobre la marcha al servidor esclavo. (Rafaelma, 2010)

Hot Standby permite acceder en modo de solo lectura a los datos disponibles en el servidor esclavo a donde se está replicando la base de datos. Se configura mediante los parámetros `hot_standby` y `max_standby_delay` en `postgresql.conf`. (rafaelma, 2010)

Hot Standby y Streaming Replication está basado en la transferencia de registros WAL entre servidores. Esta transferencia se puede realizar de registro a registro o en ficheros WAL completos. Dicha técnica se encarga de replicar absolutamente toda la base de datos y los cambios que se realicen en el servidor maestro.

¹ WAL (Write Ahead Log / REDO) para guardar toda la información sobre las transacciones y cambios realizados en la base de datos.

1.3 Análisis de soluciones existentes

La revolución científica y tecnológica en que se encuentra el mundo y Cuba está implicada en todos los campos de la sociedad a nivel internacional, por lo que las bases de datos para almacenar archivos multimedia son cada día más imprescindibles en la difusión de información en los medios. A continuación se caracterizan un grupo de soluciones que emplean bases de datos para contenido audiovisual.

1.3.1 YouTube

El sistema de Youtube es mantenido por una base de datos que almacena referencias de archivos multimedia para ser publicados en este portal de internet. Este sistema de información está dentro de las diez más grandes del mundo, con la capacidad de almacenar videos, música e imágenes. Youtube fue puesto en práctica en febrero del 2005; en marzo del 2006 contaba con más de treinta millones de videos vistos por día y en julio de ese año sobrepasó los cien millones. La plataforma fue desarrollada usando el servidor web Apache, el sistema operativo Linux (SuSe), SGBD se utilizó MySQL, Python como lenguaje de programación y un compilador Python dinámico. (Revista Ciencia e Ingeniería, 2009).

Ventajas:

- Permite realizar réplicas mediante el modelo asincrónico, donde el maestro es multi-hilo y se ejecuta en un ordenador potente capaz de manejar un gran cúmulo de información.
- La base de datos permite ampliar de forma arbitraria su estructura como elemento de escalabilidad.

1.3.2 Scribd

Scribd es una base de datos dedicada al almacenamiento de archivos multimedia con la posibilidad de referenciar presentaciones de *power point*, documentos de *Word* y *pdf*. Manejada fundamentalmente en el ámbito de la educación y la formación, en estas áreas existen campus virtuales donde básicamente se trabaja con tipos de datos multimedia. Utiliza a Oracle como SGBD e integra en un único sistema diversidad de formatos de contenido. (Revista Ciencia e Ingeniería, 2009).

Las bases de datos de YouTube y Scribd como soluciones particulares no son viables para esta investigación, puesto que su desarrollo está realizado con software propietario y uno de los principales objetivos del sistema a proponer por el autor es que sea desarrollado con software libre para contribuir a la

Capítulo 1. Fundamentación teórica de la investigación

soberanía tecnológica del país. Además necesita grandes espacios para almacenar la información y gran ancho de banda para obtener un rendimiento óptimo.

A continuación se analizarán un grupo de soluciones existentes en el ámbito nacional, en específico del departamento de Señales Digitales del centro GEYSED.

1.3.3 PRIMICIA 1.6

PRIMICIA 1.6 es una plataforma de televisión informativa la cual se encarga de proveer un canal de televisión de tipo teletexto. Dentro de los principales objetivos que persigue este producto están que se dedica a la transmisión de noticias, almacén de la información referente a estas, así como su administración y gestión. (Rosa, 2009)

1.3.4 Plataforma de Transmisión abierta para Radio y Televisión (PTARTV)

La Plataforma de Transmisión Abierta para Radio y Televisión tiene como objetivo proveer una solución tecnológica integral que permita automatizar la transmisión, administración y gestión de los canales de radio y de televisión que pudieran existir en una entidad. Desarrollado por tecnologías libres.

1.3.5 Sistema de Captura y Catalogación de Medias (SCCM)

El producto de Sistema de Captura y Catalogación de Medias tiene entre sus principales características estar desarrollado con tecnologías libres y cumplir con las siguientes funcionalidades: captura de señales de video y audio según planificaciones definidas, almacenamiento y gestión de los ficheros de video capturados en un archivo y catalogación de los materiales audiovisuales almacenados.

1.3.6 Plataforma VideoWeb

El producto VideoWeb tiene como objetivo obtener una solución de software que permita la gestión y transmisión de contenidos multimedia a través de la red de datos, por medio de un flujo constante, de manera que los usuarios no tengan que descargar los materiales hacia la PC terminal donde se quieren reproducir. Cuenta con el Subsistema de Administración y el Subsistema de Gestión y Presentación de Contenidos.

Los productos PRIMICIA, PTARTV, SCCM y VideoWeb centran su desarrollo sobre software libre, utilizando PostgreSQL como el SGBD, emplean un modelo de base de datos relacional, favoreciendo la normalización. Sus bases de datos son dinámicas con el fin de modificar y actualizar los datos. Se utiliza

Capítulo 1. Fundamentación teórica de la investigación

una técnica de replicación basada en disparadores con una distribución asíncrona, empleando el entorno Maestro-Eslavo.

Los productos nacionales analizados constituyen soluciones parciales que de ser integradas puede ser una solución viable según el objetivo planteado.

1.4 Herramientas CASE para el diseño de base de datos. Visual Paradigm

En la actualidad con el desarrollo de las tecnologías existen diferentes tipos de herramientas CASE (Computer Aided Software Engineering) para realizar el diseño de una base de datos, como es el caso de Visual Paradigm, que sirve de soporte para el modelado de UML; algunas de sus ediciones son: Enterprise, Professional, Community, Standard, Modeler y Personal, todas estas con características distintas, lo cual hace posible que el usuario seleccione la conveniente según sus necesidades. Proporciona la diagramación visual y el diseño de sus proyectos, esto ayuda a crear de manera rápida aplicaciones de calidad y a un menor costo. Facilita el código y compatibilidad hasta con diez lenguajes, permitiendo realizar tanto ingeniería inversa como directa. Proporciona la generación de base de datos y la transformación de diagramas de Entidad-Relación en tablas de ellas. (Rozic, 2004)

Además presenta la peculiaridad de crear el esquema de clases a partir de una base de datos y crear la de base de datos a partir del esquema de clases. Es una herramienta multiplataforma y útil para la generación de código fuente en PHP. Resiste múltiples usuarios trabajando sobre el mismo trabajo y ofrece la posibilidad de que los mismos perciban los cambios hechos en tiempo real.

Entre las principales ventajas que proporciona Visual Paradigm están que posee abundante documentación, tutoriales y demostraciones interactivas. Su licencia es gratuita y comercial. Es sencillo de instalar y actualizar, presentando compatibilidad entre ediciones. Por todas estas características fue la herramienta que se empleó en el trabajo de diseño de la base de datos.

1.5 Herramienta de desarrollo Pg Admin III para PostgreSQL

PgAdmin es una herramienta de código abierto para diseñar, mantener y administrar las bases de datos de PostgreSQL. Está diseñado para responder a las necesidades de los usuarios, como la escritura de consultas SQL para crear bases de datos complejas. La conexión con el servidor se puede hacer a través del protocolo TCP / IP o Unix Domain Sockets, y puede ser encriptado para una mayor seguridad. Pg Admin III está disponible en más de una docena de idiomas y es un software libre distribuido bajo la licencia de PostgreSQL. La aplicación contiene un editor SQL con resaltado de sintaxis, cuenta con un editor de código de la parte del servidor, se puede lanzar scripts programados. Es liberado con un

Capítulo 1. Fundamentación teórica de la investigación

instalador y no requiere ningún controlador adicional para comunicarse con el servidor de base de datos. (PgAdmin, 2010)

Posee importantes características como son las entradas SQL aleatorias, facilitando la interacción con el sistema de base de datos. Presenta una excelente ayuda, con pantalla de información, tablas, índices, secuencias, vistas, programas de arranque, funciones y lenguajes. Permite configurar usuarios, grupos, privilegios a través de preguntas y respuestas. Mantiene un control de revisión con mejoras de generación de script. También tiene presente la interoperabilidad, contando con ayuda para importar y exportar datos y migrar la base de datos.

Por lo anterior planteado se puede afirmar que Pg Admin III es la idónea a utilizar, como herramienta de desarrollo.

1.6 Herramienta para prueba de volumen de la base de datos. EMS Data Generator 2005 para PostgreSQL.

Las pruebas de rendimiento se utilizan para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. Además se puede aprovechar para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Permitiendo mejorar el rendimiento, enfocándose en el diseño y la arquitectura de un sistema, antes incluso del esfuerzo inicial de la codificación. (Wordpress, 2008)

EMS Data Generator para PostgreSQL es utilizado para generar datos de prueba a tablas de base de datos PostgreSQL. Brinda varios beneficios como la definición de tablas y campos para generar datos, configurar valores de rangos, generar campos de tipo carácter por máscara, cargar valores desde archivos para campos de tipo objetos binarios, así como obtener listas de valores desde consultas SQL. Además cuenta con otras características como que posee una interfaz amigable, presenta diferentes tipos de generación por cada campo, incluyendo lista, azar, generación incremental de datos, control automático sobre integridad referencial para la generación de datos a tablas vinculadas. Cuenta con una amplia variedad de generación de parámetros para cada tipo de campo. Capacidad para configurar valores nulos. Utilitario de líneas de comandos para generar datos usando el archivo que contiene la plantilla. Posibilidad de salvar los parámetros de generación, para comenzar la sesión del asistente actual. (PostgreSQL, 2008)

Capítulo 1. Fundamentación teórica de la investigación

1.7 La herramienta JMeter para la validación del diseño de la base de datos

JMeter es una herramienta de software libre desarrollada en Java, que permite realizar pruebas de carga. Su generación de carga se puede hacer de forma indirecta, mediante una aplicación y directa que se basa en la utilización de consultas grabadas en la traza o log del servidor de base de datos. Es sencillamente configurable y permite conocer los tiempos de respuesta cuando se tiene un número N de usuarios y el número real de transacciones procesadas por unidad de tiempo. Es ejecutada sobre una máquina virtual de java (JVM), generando carga para FTP, HTTP, HTTPS y SQL.

El Apache Jmeter cuenta con una interfaz gráfica de usuario, que facilita la confección de las pruebas. Brinda la posibilidad de guardar y alterar tanto los *test* desarrollados como los componentes que lo integran, lo cual trae como beneficio que se pueda reutilizar en otras nuevas. Su funcionamiento se basa en elementos integrados dentro de una estructura de árbol, cuando es de forma descendente permite determinar el orden de ejecución de los elementos dentro de la prueba. Además tiene como ventaja que el entorno de trabajo sea apropiado para ver y editar los elementos que componen un plan de prueba. Cuenta con dos entidades básicas, un plan de pruebas y las etapas o acciones que se usarán para componer las pruebas. Para lograr el funcionamiento de Apache Jmeter, se necesita tener instalado el sistema operativo Java Runtime Environment (JRE 6) y el driver postgresql-9.1-701 para la conexión a la base de datos.

1.8 Conclusiones parciales

En este capítulo fueron abordados diferentes conceptos y términos que permiten una mejor comprensión y guardan estrecha relación con la base de datos que se diseñará. Se llevó a cabo un estudio sobre los diferentes tipos de bases de datos, se clasificaron según su utilidad, necesidad o de acuerdo al escenario en el cual se aplicará, se concluyó que el modelo relacional es el más idóneo a utilizar, destacando que permite evitar la duplicidad de los registros.

Como resultado de la investigación sobre los principales SGBD (Oracle, MySQL, Microsoft SQL Server y PostgreSQL) fue seleccionado PostgreSQL por ser el más adecuado para llevar a cabo la independencia tecnológica. Se profundizó en el trabajo con las réplicas, llegándose a la conclusión de que se creará un entorno de replicación maestro-esclavo con un modelo de distribución asíncrono y con la técnica de replicación basada en disparadores. Se examinaron diversas herramientas útiles para el diseño de la base de datos seleccionándose Visual Paradigm para el modelado, el PgAdmin para mantener y administrar la base de datos. Se decidió utilizar para la validación funcional de la base de datos la herramienta Jmeter.

Capítulo 1. Fundamentación teórica de la investigación

Se planteó que las soluciones existentes analizadas, cumplen parcialmente con las características que requieren la gestión y transmisión de contenidos audiovisuales en un sistema de este tipo.

Descripción y análisis de la solución propuesta

2.1 Introducción

En este capítulo se abordará sobre los principales procesos que se llevaron a cabo a la hora de diseñar e implementar la base de datos para la plataforma de transmisión de contenidos multimedia. Se hace una fundamentación de la arquitectura a utilizar, se explican los requisitos funcionales y no funcionales, se describen las clases persistentes y se muestra el diagrama lógico de datos o diagrama de clases persistentes con la descripción de las tablas de la base de datos a diseñar. Además se describe como realizar la optimización de la base de datos.

2.2 Descripción de la arquitectura

La arquitectura de software es esencial para el desarrollo de un sistema, esta indica la estructura y funcionamiento e interacción entre las partes de este. La arquitectura de sistemas de bases de datos de tres esquemas fue aprobado por la ANSI-SPARC (*American National Standard Institute - Standards Planning and Requirements Committee*) en 1975 con el objetivo de separar los programas de aplicación de la base de datos física. Para el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En esta arquitectura, los esquemas se pueden definir en los tres niveles siguientes: (Andrés, 2001)

- Nivel interno: posee un esquema interno que se encarga de describir la estructura física de almacenamiento de base de datos. Empleando un modelo físico de datos y los únicos datos que existen están realmente en este nivel. (Andrés, 2001)
- Nivel conceptual: posee un esquema conceptual describiendo la estructura de toda la base de datos. Además oculta los detalles físicos de almacenamiento y trabaja con elementos lógicos como entidades, atributos y relaciones. (Andrés, 2001)
- Nivel externo o de vistas: cuenta con varios esquemas externos o vistas de usuario. Cada uno describe la visión que tiene de la base de datos a un grupo de usuarios, ocultando el resto. (Andrés, 2001)

Capítulo 2. Descripción y análisis de la solución propuesta.

Soporte arquitectónico

En la base de datos desarrollada para el sistema de gestión y transmisión de contenidos audiovisuales se utilizó un esquema de trabajo implementado por el gestor de base de datos PostgreSQL. Este esquema denominado public (público) es el encargado de almacenar toda la información que se genera en las cuatro bases de datos integradas, sumando 185 tablas que conforman la base de datos final. No se implementaron procedimientos almacenados previendo la posibilidad de un cambio del SGBD lo que conllevaría a una pérdida de las funciones de dichos procedimientos. Además, la migración de la base de datos no es directa. Existen herramientas públicas que traducen al 80% de efectividad por lo que no es recomendable. También se puede traducir manualmente pero el trabajo sería engorroso y lento, ya que se migrarían de 20 a 30 procedimientos diarios por hombre. Por tanto se definió la capa lógica de la aplicación como responsable de la entrada y salida de los datos.

Estándares de Nomenclatura

La idea central de definir un estándar de nomenclatura es lograr que en la notación de la base de datos se cumpla con tres conceptos importantes: consistencia, legibilidad y documentación. Las normas utilizadas son:

- Se determina que el nombre de la base de datos se escriba con letras en minúsculas. En caso de ser un nombre compuesto las palabras se separan por “_”. Existen dos entornos que interactúan con el servidor de base de datos. El primero y más importante es el servidor maestro llamado “master” y el segundo el “esclavo”.
- El nombre a emplear para las tablas se escribe igualmente con letras en minúsculas, sin acentos y las palabras separadas por el símbolo “_”. Ejemplo: archivo_multimedia.
- El nombre a emplear para los campos debe ser intuitivo y escribirse con letras en minúsculas, comenzando por el prefijo correspondiente y las palabras separadas por el símbolo “_”. Si el campo es un identificador de llave primaria debe ser **id**, en caso de que no exista otro atributo único que pueda identificar a la tabla. Si es un identificador de llave foránea debe estar constituido por **id** y el nombre de la tabla de la cual se recibe la llave, separados por “_”. Ejemplo: id_ficha. Si la tabla tiene llave primaria entonces la llave foránea se compone de nombre “_” y el identificador. Ejemplo: solicitud_id.
- Cuando exista una relación de mucho a mucho, al crearse la nueva tabla se renombrará la misma con los nombres de las tablas que se relacionan separadas por “_” Ejemplo: solicitud_ficha.

Capítulo 2. Descripción y análisis de la solución propuesta.

2.3 Requisitos funcionales y no funcionales

Rogers Pressman plantea: “La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente”. (Pressman, 2010)

Los requisitos engloban las necesidades o las condiciones a satisfacer para resolver un problema o lograr un objetivo en cualquier software. A continuación se relacionan los requisitos funcionales y no funcionales capturados para la base de datos del sistema de gestión y transmisión de contenidos audiovisuales.

Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, se encargan de especificar acciones que el sistema debe ser capaz de realizar. (Wieggers, 2003). A continuación se representan los requisitos funcionales más significativos del sistema.

R1 Adicionar tipología de archivo multimedia: la base de datos debe ofrecer la posibilidad de almacenar la información referente a una tipología de archivo multimedia

R2 Adicionar campos a una tipología existente: la base de datos debe permitir la adición de nuevos campos o propiedades de una tipología asociada.

R3 Editar tipología de archivo multimedia: el contenedor de información del sistema debe permitir que la información almacenada pueda ser modificada o eliminada.

R4 Crear tipología previamente adicionada: la base de datos debe permitir crear físicamente la tabla de la tipología, la tipología queda disponible para ser utilizada a la hora de adicionar archivos multimedia.

R5 Adicionar archivos multimedia: la base de datos debe permitir almacenar los archivos multimedia en el servidor de medias y crear las referencias pertinentes en la base de datos con todos los datos asociados a estos.

R6 Modificar datos asociados a los archivos multimedia: la base de datos debe permitir modificar los datos asociados a los archivos multimedia.

R7 Eliminar archivos multimedia: la base de datos debe permitir eliminar archivos multimedia con los datos asociados a estos.

Capítulo 2. Descripción y análisis de la solución propuesta.

R8 Crear publicación multimedia: la base de datos debe permitir crear publicaciones multimedia a partir de los archivos multimedia previamente almacenados en el servidor de medias, permitiendo predefinir el tiempo de publicación acotado por un inicio y fin.

R9 Modificar publicación multimedia: la base de datos debe permitir modificar los datos de la publicación multimedia.

R10 Eliminar publicación multimedia: la base de datos debe permitir eliminar publicaciones multimedia.

R11 Publicar archivo multimedia: la base de datos debe permitir activar las publicaciones multimedia previamente elaboradas, creando un enlace simbólico al archivo en el punto de publicación.

R12 Terminar publicación archivo multimedia: la base de datos debe permitir desactivar las publicaciones multimedia previamente publicadas, eliminando el enlace simbólico del punto de publicación.

R13 Realizar búsqueda de los usuarios registrados: la base de datos debe permitir realizar búsquedas de los usuarios registrados.

R14 Realizar búsqueda de publicaciones de archivos multimedia: la base de datos debe permitir realizar búsquedas de publicaciones de archivos multimedia.

R15 Crear señal en vivo: la base de datos debe permitir crear una señal en vivo.

R16 Modificar datos de señal de audio o video: la base de datos debe permitir modificar los datos asociados a las señales de audio o video en vivo.

R17 Eliminar señal de audio o video en vivo: la base de datos debe permitir eliminar señales de audio o video en vivo.

Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe tener. Contienen características que hacen al producto atractivo, usable, rápido y confiable.

Confiabilidad:

- El SGBD debe poseer propiedades para la administración de roles y usuarios restringiendo el acceso a los datos.

Rendimiento:

- Al realizar cualquier gestión de la información en la base de datos, ya sea mediante acciones de inserción, búsqueda, eliminación o modificación de datos, se debe recibir la respuesta de la acción realizada en un período de 0.1 a 0.8 segundos.

Seguridad:

Capítulo 2. Descripción y análisis de la solución propuesta.

- Se deberán realizar copias de respaldo que consigan restaurar el sistema en caso de cualquier fallo crítico o pérdida total de la información.
- Según el privilegio que tenga el usuario podrá acceder a las opciones permitidas.

Fiabilidad:

- El sistema debe estar disponible las 24 horas.
- El servidor de base de datos debe mantener buenas prestaciones ante la demanda concurrente de peticiones.

Eficiencia:

- El servidor de base de datos debe ser capaz de manejar más de 700 conexiones simultáneas.

Software:

- El SGBD a utilizar debe ser: PostgreSQL 9.1.

Hardware:

Los requerimientos mínimos para que el servidor de base de datos funcione correctamente son:

- Tarjeta de Red: Más 100 MB/s
- Procesador: Intel Core2 Dúo a 2.00 GHZ o superior
- RAM: 1 GB o más
- Disco duro: 160 GB o más

2.4 Diseño de la base de datos

Es importante asegurar que la base de datos está correctamente diseñada para que tenga eficiencia y se pueda seguir utilizando. El diseño puede ser complejo, pero con algunas reglas será mucho más fácil crear una base de datos perfecta. Para realizar un buen diseño es necesario evitar la redundancia y garantizar que la información sea correcta y completa. Algunas consideraciones para hacer el diseño de la base de datos son:

- La velocidad de acceso.
- El tamaño de la información.
- El tipo de la información.
- El acceso a la información.

Capítulo 2. Descripción y análisis de la solución propuesta.

- El comportamiento del manejador de la base de datos con cada tipo de información.

2.4.1 Modelo lógico y físico de la base de datos

La base de datos maneja un gran cúmulo de información además, la seguridad del sistema y sus funcionalidades tienen una importancia capital. En la figura 2 se muestra el diagrama de clases persistentes, que permite observar las relaciones y características de las 185 entidades.

Capítulo 2. Descripción y análisis de la solución propuesta.

2.4.2 Descripción de las tablas

Las tablas de la 1 a la 9 representan las descripciones de las principales tablas de la base de datos, las demás se encuentran en el anexo #1.

Nombre: archivo_multimedia	
Tipo de clase: entidad	
Atributo	Tipo
Id	Bigint
Nombre	Text
Ubicación	Text
disponible	Boolean
archivo_multimedia_genero_id	Integer
created_at	timestamp(6) without time zone
updated_at	timestamp(6) without time zone
ftp_punto_publicacion_id	Integer
id_tipologia_am	Integer
nombre_fichero	Text
autor_id	Integer
id_dato_tipologia_am	Integer
Duracion	time without time zone
Uid	Integer
Catalogada	Boolean
Fotogramaa	Boolean
Fecha	Date

Capítulo 2. Descripción y análisis de la solución propuesta.

Streaming	Boolean
Procedencia	Text
Id_restriccion	Integer
Codigo	Integer

Tabla 1: Archivo_multimedia

Nombre: archivo_multimedia_genero	
Tipo de clase: entidad	
Atributo	Tipo
Id	Bigint
Nombre	Text
created_at	timestamp without time zone
updated_at	timestamp without time zone

Tabla 2: Archivo_multimedia_genero

Nombre: archivo_multimedia_lista	
Tipo de clase: entidad	
Atributo	Tipo
Id	Bigint
nombre_lista	character varying(100)
Uid	int_unsigned
Peso	int_unsigned

Tabla 3: Archivo_multimedia_lista

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: archivo_multimedia_lista_node	
Tipo de clase: entidad	
Atributo	Tipo
Lid	int_unsigned
Nid	int_unsigned
Peso	int_unsigned

Tabla 4: Archivo_multimedia_lista_node

Nombre: canal	
Tipo de clase: entidad	
Atributo	Tipo
id_canal	Bigint
Nombre	character varying(255)
Puerto	character varying(255)
dir_ip	character varying(255)
Estado	Integer
Descripcion	character varying(255)
url_logo	character varying
Activo	Boolean
Protocolo	character varying(255)

Tabla 5: Canal

Nombre: canal_espacio_televisivo	
Tipo de clase: entidad	

Capítulo 2. Descripción y análisis de la solución propuesta.

Atributo	Tipo
id_canal	Integer
id_espacio	Integer
id_canal_espacio_televisivo	Bigint

Tabla 6: Canal_espacio_televisivo

Nombre: espacio_dia	
Tipo de clase: entidad	
Atributo	Tipo
id_espacio_dia	Bigint
Id_espacio_televisivo	Bigint
Id_dia	Bigint

Tabla 7: Espacio_dia

Nombre: espacio_televisivo	
Tipo de clase: entidad	
Atributo	Tipo
id_espacio_televisivo	Bigint
hora_inicio	time(6) without time zone
hora_fin	time(6) without time zone
Nombre	Character varying(255)
fecha_espacio	Date

Tabla 8: Espacio_televisivo

Nombre: espacio_televisivo_archivo_multimedia	
Tipo de clase: entidad	

Capítulo 2. Descripción y análisis de la solución propuesta.

Atributo	Tipo
Orden	Integer
Fecha	Date
id_espacio_televisivo	Integer
id_media	Integer
id_canal	Integer
hora_inicio	time without time zone
id_epacio_media	Integer

Tabla 9: Espacio_televisivo_archivo_multimedia

2.5 Optimización de la base de datos

La optimización de la base de datos permite mejorar el rendimiento y funcionamiento, de acuerdo a sus características y propósitos. La optimización se puede lograr de dos formas, trabajando tanto en la configuración donde se trata la parte concerniente al hardware y los archivos de configuración o implementando ajustes algebraicos a las consultas que están comprendidas por los índices, *trigger*. A continuación se mostrarán algunos elementos fundamentales a tener en cuenta a la hora de optimizar PostgreSQL.

Optimización para PostgreSQL

Hardware: se manejan elementos como el CPU que es el encargado de realizar los cálculos matemáticos en una computadora, mientras mejor sea su rendimiento permitirá mayor velocidad de acceso. La memoria RAM mientras mayor sea su capacidad, más programas podrá conmutar simultáneamente, esto permitirá minimizar el acceso a disco que es cientos de veces más lento que el acceso a la RAM. (Hall, 2008)

Postgresql.conf: en este fichero se encuentran los parámetros que determinan el entorno de ejecución del servidor de Postgres, algunos de ellos se pueden fijar mediante las opciones de arranque del servidor y tienen valores por defecto, algunos vinculados al código fuente y otros como una opción de configuración en tiempo de compilación. En este elemento se optimizan parámetros como la caché, la

Capítulo 2. Descripción y análisis de la solución propuesta.

memoria compartida y la paginación. En PostgreSQL estos aspectos son vistos de la siguiente forma: `shared_buffers`, `cache_miss`, `wal_buffers`, `full_page_writes`, `effective_cache_size`. (Hall, 2008)

Optimización del diseño de los datos y la aplicación: este aspecto trata de optimizar el diseño de las tablas de la base de datos, la forma de indexar, el diseño de las consultas teniendo en cuenta que los datos deben ser persistentes, relaciones y atributos más importantes, y estructurar toda la información de forma tal que cumpla con los requisitos de la base de datos. Después se procede a normalizar los datos para evitar la redundancia. En el proceso de diseño de los datos se debe tener en cuenta cuales atributos se indexan y cuáles no, teniendo en cuenta los campos de mayor frecuencia de acceso y los que se utilizan comúnmente en consultas de comparación. Además la aplicación que accede a sus datos debe intentar minimizar la cantidad de conexiones y transacciones en la medida de lo posible y realizarlas correctamente. (Hall, 2008).

Afinamiento de consultas: este aspecto se encarga de optimizar las consultas SQL realizadas en la base de datos. Una consulta en SQL puede ser escrita de varias formas y esto trae consigo que son ejecutadas de formas diferentes, por lo que un planificador de consultas de bases de datos decidirá cuál es la manera más eficiente de ejecutar una consulta y hacerlo lo suficientemente rápido.

Por último el optimizador debe garantizar un plan de ejecución que logre la certeza de la consulta y en su comprobación tomar un valor significativo durante las pruebas mediante el “`explain analyze`” para conocer el camino de la ejecución de las consultas, los tiempos asociados a la misma y mejorarlas en caso de que los resultados no respondan a los requisitos. (Hall, 2008)

Dentro de los ajustes algebraicos de la consulta se encuentran los índices, estos son un subconjunto ordenado de las columnas de una tabla, con una entrada que apunta a la tupla correspondiente, en la tabla `archivo_multimedia_lista` de la base de datos se presenta un ejemplo vigente como es: `idxes: archivo_multimedia_lista_id_idx`. Otro aspecto muy importante a la hora de optimizar son los desencadenadores o *triggers*, estos son funciones que se ejecutan de forma automática en respuesta a ciertos eventos que ocurren en la base de datos, ejemplo de ello ocurre en la tabla `canal` y `canal_espacio_televisivo` donde se emplean el *trigger* `canal_tr` y `canal_espacio_televisivo_tr` con el objetivo de notificar cambios en las planificaciones de transmisión de esas tablas. Los *triggers* `notificar_cambio_canal`

Capítulo 2. Descripción y análisis de la solución propuesta.

y mcabio_canal utilizan la función listen/notify² encargada de enviar una señal a la aplicación de los cambios realizados en esas tablas por conceptos de actualización, eliminación e inserción de datos. Por último la tabla espacio_televisivo_archivo_multimedia presenta el *trigger*: archivo_multimedia_tr, que tiene como función notificar a la aplicación de un cambio realizado en los espacios televisivos en el día actual.

2.6 Conclusiones parciales

En este capítulo se presentó el resultado del diseño de la base de datos, este fue alcanzado a partir de la necesidad descrita por los requisitos funcionales y no funcionales. Mediante el diagrama de clases persistentes, sus interrelaciones y la descripción de las tablas más importantes de la base de datos diseñada, se obtuvo un contenedor de información capaz de permitir la gestión y manipulación de la información necesaria en un sistema para la gestión y transmisión de contenidos audiovisuales.

La descripción de la arquitectura de la base de datos diseñada permitió darle soporte y seguimiento a la propuesta, garantizando así la continuidad o escalabilidad de la base de datos. Se tuvo en cuenta en el diseño de la base de datos, elementos de optimización tanto de hardware como de software, reflejado en la optimización de consultas SQL.

² listen/notify proporcionan una forma simple de la señal o un mecanismo de comunicación entre procesos para una colección de procesos que acceden a la misma PostgreSQL de la base de datos.

Capítulo 3

Validación de la Propuesta

3.1 Introducción

En este capítulo se aborda el proceso de validación de la base de datos para la plataforma de transmisión de contenidos multimedia. Está compuesta por una validación teórica y una funcional donde se hace referencia a aspectos como son la integridad, la seguridad, la redundancia de la información y la normalización de la base de datos; a la cual se le realizaron pruebas para comprobar su eficiencia y buen funcionamiento.

3.2 Integridad de los datos

La integridad como concepto proviene del término latino *integritas* y hace referencia a la cualidad del íntegro. El íntegro se refiere a cuando no carece de ningunas de sus partes considerado como intachable. (Wordpress, 2008)

La integridad en una base de datos significa la corrección y exactitud de la información almacenada, es la encargada de verificar que los datos tengan calidad. Se puede definir el término de integridad en una base de datos como un conjunto de seguridades que hacen que los datos sean correctos. (Wordpress, 2008)

Al utilizar sentencias como INSERT, DELETE o UPDATE, la integridad de los datos puede verse afectada, ya que esto posibilita que se puedan añadir datos no válidos o modificarse datos existentes en forma incorrecta, imposibilitando el cumplimiento de la integridad. (Wordpress, 2008)

Al definir la estructura de datos del modelo relacional, se determina el modelo conceptual, se procede a aplicar las reglas de integridad de los datos almacenados en dicha estructura, cumpliendo con ciertos parámetros asociados a dicha estructura, de igual forma, la descripción de un atributo sobre un dominio determinado impone una restricción, la cual es llamada restricción de dominio. Existen dos reglas fundamentales que son restricciones que debe cumplir una base de datos: la regla de integridad de entidades y la regla de integridad referencial. Para un mejor entendimiento es necesario conocer los conceptos de nulo y dominio. (Savendra, 2009)

Capítulo 3. Validación de la propuesta.

Nulo: Es un indicador que indica que el dato falta, no es aplicable o es vacío. Por conveniencia, un dato que falta normalmente se dice que tiene valor nulo, sin embargo este último no es un valor de dato real. (Savedra, 2009)

Dominio: Se refiere a posibles valores que puede tener un campo, no es más que un tipo de dato definido por el sistema o por el usuario. Esta característica, usada en forma correcta, se convierte en mecanismo de control, restricción y validación de los datos a ingresar. (Savedra, 2009)

3.3 Integridad de entidad

Las restricciones de entidades aseguran la integridad que son modeladas por el sistema, es decir la existencia de una clave principal es una restricción de entidad que impone la regla de que cada entidad debe estar identificada de forma única; por lo que no está permitido que algún componente de la clave primaria acepte valores nulo. Esta también define una fila como entidad única para una tabla determinada, además exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones únicas, o restricciones de llaves primarias. (Savedra, 2009)

Esta regla define lo siguiente: (Savedra, 2009)

- Las tuplas en las relaciones base representan entidades en la realidad.
- Las entidades en la realidad son identificables por definición.
- Sus contrapartes en la base de datos también deben ser identificables.
- Los valores de la clave primaria sirven como identificadores en la base de datos.
- Los valores de clave primaria no pueden ser nulos.

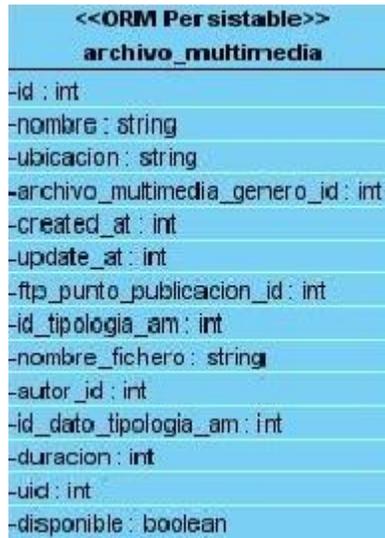
En la base de datos de la plataforma de transmisión de contenidos multimedia cada entidad representada en el modelo relacional tiene una única llave primaria y su valor es no nulo y numérico.

3.4 Integridad de dominio

La integridad de dominio se encarga de garantizar la validez de las entradas para una columna determinada, restringe el tipo de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones de llaves foráneas, restricciones CHECK, definiciones por defecto, definiciones no nulas y reglas. (Server C.d.D.d.S, 2009)

Capítulo 3. Validación de la propuesta.

La base de datos refleja la integridad de dominio en diferentes etapas, ejemplo de esto se tiene en la entidad `archivo_multimedia`, su atributo `disponible` es de tipo de dato *booleano* que se encuentra en la figura 3, restringiéndolo a que sea de tipo verdadero o falso.



```
<<ORM Persistable>>
archivo_multimedia
-id : int
-nombre : string
-ubicacion : string
-archivo_multimedia_genero_id : int
-created_at : int
-update_at : int
-ftp_punto_publicacion_id : int
-id_tipologia_am : int
-nombre_fichero : string
-autor_id : int
-id_datos_tipologia_am : int
-duracion : int
-uid : int
-disponible : boolean
```

Figura 3: Integridad dominio

3.5 Integridad referencial

La integridad referencial en las bases de datos es la encargada de garantizar que una entidad siempre esté relacionada con otras entidades válidas, supone que todos los datos sean correctos, sin repeticiones, datos perdidos o relaciones mal resueltas. (Wordpress, 2008). La condición principal de integridad referencial exige que no existan en el modelo relacional valores para las claves foráneas sin coincidencia con los valores de las claves primarias asociadas a ella. (Hall)

La integridad referencial responde a las siguientes reglas:

- No se podrá introducir un valor en la tabla relacionada si antes no ha sido introducido en la tabla principal.
- No se puede eliminar un registro de una tabla principal si existen registros coincidentes en la tabla relacionada.

- No se puede cambiar un valor de la clave primaria en la tabla principal si el registro tiene registros relacionados.
- Integridad definida por el usuario.

Para hacer más explícito esto, se tiene el siguiente ejemplo: en la figura 4, se representan las relaciones entre Solicitud_Cliente y Cliente.

El modelo no podría tener registrado una solicitud para un cliente inexistente, ya que no se cumpliría la condición de integridad referencial. Solo es válido realizar una solicitud a un cliente existente y no tendría sentido generar una solicitud a un cliente inexistente a no ser que se deseara realizar una solicitud falsa y no es objetivo permitir que esto suceda. Este tipo de consistencia de la información es de lo que se ocupa la integridad referencial y todos los motores de datos permiten definirla, aunque no de forma automática.

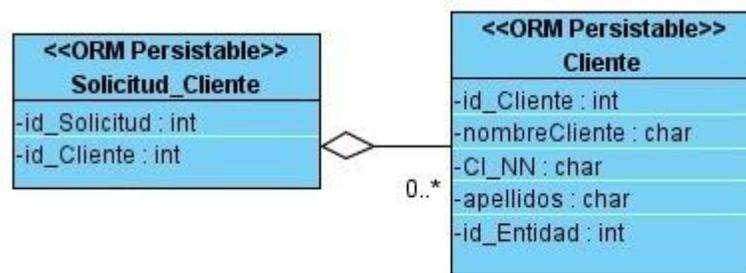


Figura 4: Integridad referencial

3.6 Integridad semántica

La integridad semántica es la propiedad que tiene como restricción que se debe modelar la base de datos tal y como se describe. Las restricciones se definen en el esquema lógico y el SGBD debe velar por su cumplimiento. La comprobación se realiza cuando en la base de datos se ejecuta una operación de actualización.

Restricción de integridad estática, se deben cumplir en cada estado de la base de datos:

- Sobre dominios: de valor.
- Sobre atributos: valor no nulo, de rango.
- Sobre relaciones: clave primaria y claves ajenas.

- Sobre la base de datos: condiciones de búsqueda general.
- Restricción de integridad de transición: se debe cumplir en dos estados consecutivos.

El mantenimiento de la integridad semántica se encarga de evitar que se inserten en la base de datos la información que represente un estado no permitido del universo. Se refleja en los procedimientos de comprobación de la integridad mediante reglas de actividad y *triggers*.

3.7 Normalización de la base de datos

Un elemento fundamental en las bases de datos es la normalización, esta característica plantea como evitar la redundancia y el fácil acceso y actualización de la información contenida. La normalización trata de transformar los datos complejos en un conjunto de pequeñas estructuras de datos y hace factible su manipulación a la hora de mantenerlos. Según Edgar Frank Codd, la normalización se define en normas, a las que llamó formas normales (FN), las cuales están enumeradas. Una relación para que esté en una FN antes debe estar en todas las FN que la anteceden. (Rozic, 2004)

El proceso de Normalización cuenta con las siguientes formas normales:

- Primera Forma Normal (1FN).
- Segunda Forma Normal (2FN).
- Tercera Forma Normal (3FN)
- Forma Normal de Boyce Codd.
- Cuarta Forma Normal (4FN).
- Quinta Forma Normal (5FN).

Una base de datos está en su 1FN mientras sus relaciones no contengan grupos repetitivos y se eliminen los atributos multivaluados, compuestos y sus combinaciones. La 2FN garantiza que se controle la mayoría de los problemas de lógica, se insertan registros sin una abundancia de datos en la totalidad de las tablas, provocando que se eviten las anomalías a la hora de realizar actualizaciones. La 3FN garantiza que no haya dependencias transitivas, así se evitan los errores de lógica cuando se insertan o eliminan registros. (Rozic, 2004)

La FN de Boyce Codd se utiliza cuando no se satisface totalmente los criterios de normalización, en los casos que la relación posee más de una clave candidata; mientras que la 4FN se aplica para dependencias multivaluadas y la 5FN cuando no se satisface que toda dependencia de reunión es consecuencia de las claves candidatas de la relación. (Rozic, 2004)

3.8 Análisis de la redundancia de la información

La redundancia de la información hace referencia al almacenamiento de los datos varias veces en diferentes lugares, provocando problemas como:

- El incremento del trabajo: ocurre cuando los datos que están almacenados en más de un lugar se actualizan y requieren hacer el mismo trabajo varias veces.
- La pérdida de capacidad de almacenamiento: ocurre cuando los datos que están almacenado en lugares distintos, ocupan más espacio del medio de almacenamiento.
- La inconsistencia de los datos: ocurre cuando se actualiza los datos en un lugar del sistema pero el duplicado no se actualiza.

Con la realización del diseño de la base de datos se logró evitar la aparición de información repetida. Después de llevar la base de datos hasta la 3FN, queda libre de redundancias, ya que se elimina completamente la presencia de datos repetidos innecesariamente. La normalización es una técnica que elimina en gran medida esta dificultad, aunque en base de datos grandes es imposible eliminarla al ciento por ciento.

3.9 Análisis de la seguridad y recuperación

La seguridad es un factor crítico a tener en cuenta durante el desarrollo de la base de datos, la cual tiene como principal objetivo proteger la información de usuarios no autorizados para garantizar que dichos datos cuenten con un gran por ciento de protección. Esto trae como beneficios evitar posibles ataques e impedir cualquier acceso no autorizado. Se definen para cada usuario sus privilegios, garantizando así que ninguno por error o intencionalmente pueda eliminar, modificar, adicionar, copiar, usar o difundir la información almacenada en la base de datos sin tener los permisos necesarios para realizar estas operaciones.

El SGBD PostgreSQL garantiza el control de acceso por parte de los usuarios a la base de datos, mediante los privilegios que tengan concedidos y los lugares donde ellos se encuentren ubicados. Permite configurar los permisos de conexión para los *host* y los usuarios de la base de datos utilizando las opciones del fichero de configuración del SGBD Postgresql "pg_hba.conf", definiéndose en dicho fichero qué direcciones de IP tendrán acceso y en qué modo se conectarán. Además se efectúan las configuraciones pertinentes en el fichero "pg_ident.conf", para las conexiones entre los usuarios y las bases de datos, definiendo a qué base de datos se pueden conectar. También da la opción de configurar

Capítulo 3. Validación de la propuesta.

a qué tablas se pueden acceder, esto lo realiza mediante las sentencias del DCL, GRANT para otorgar permisos y REVOKE para eliminar los permisos otorgados.

Para mayor seguridad en la base de datos se cifra la contraseña utilizando el algoritmo de reducción criptográfico de 128 bits llamado MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5). Este algoritmo posee las características de imposibilitar la reconstrucción de la cadena original a partir del resultado y de encontrar dos cadenas de texto que generen el mismo. En el punto de destino, para comprobar si la contraseña es correcta, se cifra de igual manera y se comparan las formas cifradas. Se define una clave para el usuario postgres, luego se procede a configurar el archivo pg_hba.conf para que PostgreSQL valide la contraseña con el algoritmo MD5.

Para garantizar la consistencia y la recuperación de sistemas ante fallos, se hace necesario definir una serie de pasos o guías que debe cumplir el administrador de la base de datos de un sistema informático, es por ello que se define para el desarrollo actual una política de salva y recuperación de datos ante fallos. Según la política de seguridad informática de la UCI para la administración y mantenimiento de bases de datos, los administradores de las mismas deben crear copias de respaldo y salvadas de la información de los sistemas operativos. Las aplicaciones centrales que sustentan los diferentes servicios informáticos de la comunidad universitaria tienen el objetivo de garantizar la recuperación ante cualquier contingencia. Los medios de respaldo deberán probarse regularmente y verificar su estado de actualización con el fin de asegurar que pueda confiarse en ellos para uso de emergencia cuando sea necesario.

Para la seguridad de los datos se cuenta con un servidor y otro ordenador, ambos en la misma subred, esta última, con el fin de proteger la base de datos en caso de adversidad. Las salvadas se pueden realizar de las siguientes formas:

- Compactado del directorio de datos: se detiene el servicio de PostgreSQL. Seguidamente se realiza la copia en *.rar o *.zip del directorio /data/ de la carpeta de instalación de PostgreSQL. Se inicia el servicio de PostgreSQL. Finalmente se copia el *.rar o *.zip en algún soporte legible como es el caso de un disco duro o memoria flash. Para recuperar la copia se descomprime el directorio /data/ de la carpeta de instalación de PostgreSQL.
- Salvadas automáticas: PostgreSQL las permite realizar a petición del cliente, para ello utiliza el comando pg_dump el cual indica al SGBD que va a realizar la salva de la base de datos. Se le provee los parámetros: nombre de la base de datos, usuario y contraseña para conectarse a la misma. Para restaurar la salva, se utiliza el comando pg_restore, indicándole la salva que se desea restaurar.

Capítulo 3. Validación de la propuesta.

Como la base de datos es relativamente grande y por ser dinámica, tiende a aumentar la capacidad de las salvas que se van a realizar, cada vez que ocurra algún cambio en el contenedor de información, haciéndose manualmente una salva o *backup* del sistema evitando cargar al servidor de recurso. La recuperación en caso de fallo se realizará a partir del último fichero de salva que haya realizado el sistema a la base de datos, a partir de una estructura como la que se muestra y se propone a continuación: nombre_fecha_hora.backup. Por ejemplo Plataforma_21/2/2012.backup.

3.10 Análisis de la replicación de datos.

En el capítulo uno se abordó el tema de la replicación, la cual haciendo uso de la red se encarga de realizar copias de la base de datos, posibilitando su sincronización. Es fundamental para mejorar el funcionamiento y garantizar la disponibilidad porque posibilita variantes alternas en el acceso a los datos. Además, la aplicación puede continuar funcionando en caso de que el servidor principal experimente una falla, debido a la existencia de los otros servidores con datos replicados, los cuales continuarán siendo accesibles. Los procesos de réplica se realizan en su mayoría entre bases de datos con estructuras idénticas. Entre las diferentes ventajas que brinda la replicación son:

- Provoca que se elimine el impacto de la caída de la red.
- Proporciona entornos para balanceo de carga.
- Aumenta la disponibilidad de la información.
- Proporciona copias de seguridad ante desastres, caídas o fallos.

Se manipula el entorno de réplica mediante las técnicas Hot Standby y Streaming Replication. Ellas están implementadas en el núcleo de PostgreSQL y contienen lo necesario para instalar un sistema de replicación maestro_esclavo. Esta última tiene la particularidad de ser de solo de lectura, facilita a un solo maestro recibir consultas de lectura/escritura, mientras los esclavos son solo encargados de aceptar consultas de lectura. Esto garantiza disminuir el trabajo del servidor principal en lo referente a consultas de solo lectura, posteriormente se procede a instalar dos servidores similares con el sistema operativo Ubuntu 12.04 y PostgreSQL 9.1. Finalmente el sistema replica toda la base de datos y los cambios que se realizan en el servidor maestro. Para la configuración y aspectos a tener en cuenta, se anexa a esta investigación el manual de configuración confeccionado específicamente para esta base de datos.

3.11 Validación de la base de datos

Las pruebas permiten validar las bases de datos. Este no es un proceso complejo y sí necesario, por lo cual el equipo de calidad y de desarrollo necesita realizarlas, contando con ventajosas características:

- **Confiable:** realizan exactamente las mismas operaciones cada vez que se ejecutan, evitando en gran medida el error humano.
- **Repetible:** se observa cómo reacciona la base de datos bajo ejecución repetida.
- **Programable:** son programadas de forma sofisticada y automatizada.
- **Entendible:** brindan la posibilidad de construir un test de pruebas que cubra cada característica del uso de la base de datos.
- **Reutilizable:** se pueden reutilizar las pruebas en otras nuevas, aun realizando cambios en la interfaz de usuario.
- **Rápido:** los usuarios automatizados son rápidos, por tanto el funcionamiento de las pruebas en herramientas se realiza de igual manera.
- **Reducción de costos:** los recursos para las pruebas se reduce. Las inversiones asociadas disminuyen.

Una base de datos puede ser validada si se hace uso de pruebas de volumen, carga y estrés. El objetivo de estas, es certificar el diseño de la base de datos, así como las consultas existentes en la misma, sirviendo de apoyo al programador que trabajará con ella.

Las pruebas de volumen: permiten analizar el comportamiento de la base de datos con volúmenes de datos almacenados similares a los esperados en la explotación real del sistema. La base de datos se pobló con datos aleatorios generados por una herramienta llamada EMS Data Generator, esta herramienta proporciona una cantidad considerable de datos, llenados por el propio administrador. Se configuró teniendo en cuenta los tipos y volúmenes, con los datos reales que maneja un sistema de gestión y transmisión de contenidos audiovisuales. Se concluye que este generador se pudiera considerar una prueba más, ya que si existe inconsistencia en el diseño de la base de datos este no comienza el poblado de datos hasta tanto no quede un correcto diseño. Por los resultados obtenidos se puede afirmar que:

Capítulo 3. Validación de la propuesta.

- No se presentaron problemas de límite de capacidad, ni de volumen de datos.
- No presentó desbordamientos de matrices, columnas, atributos, tipos de datos, ni peticiones excesivas de memoria.
- Las llaves autogeneradas se encontraban en el rango especificado.
- Tampoco existió problemas con los tipos de datos definidos en el diseño.

Las pruebas de cargas: son empleadas para observar el comportamiento de la base de dato bajo cierta cantidad de peticiones. Esta carga se puede dar por el número esperado de usuarios concurrentes en la base de datos y que realizan un número específico de transacciones durante el tiempo que dura la carga. Se obtiene como resultado los tiempos de respuesta de todas las transacciones. Es fundamental emplearla porque el sistema soporta un gran volumen de usuarios y transacciones concurrentes. La ejecución de estas pruebas se realizó en una estación cliente con el Apache Jmeter configurado directamente con el servidor de base de datos. Este último está soportado por un sistema operativo Ubuntu en su versión 11.10, con un microprocesador: Intel Pentium IV a una velocidad a 2.6 GHz y una memoria ram: 1.0 GB.

La configuración del Apache Jmeter se realizó de la siguiente forma: se seleccionó el plan de pruebas (Test Plan) y se procedió a añadir para luego seleccionar, el elemento que se está buscando: Grupo de Hilos (Thread Group), donde se definieron la cantidad de usuarios que realizarían peticiones al servidor. Seguidamente se configuró el acceso a la base de datos en el archivo configuración de la conexión JDBC, en el cual se detalla el nombre de la variable que será usada para las peticiones, la dirección URL mediante la que se conectan los usuarios, definiéndose los parámetros de *login*. Luego de configurada la conexión se pasó a definir la petición que se realizaría a la base de datos en el archivo petición JDBC. Después se añade un *listener* que recoja los datos y otro para mostrar el gráfico de resultados. Una vez terminada la configuración se ejecuta la prueba.

Se tuvieron en cuenta las siguientes variables:

- Media: tiempo promedio de respuesta en milisegundos de todas las peticiones.
- Mediana: tiempo promedio en milisegundos de la mitad de los resultados, la otra mitad tomara un tiempo entre la mediana y el valor máximo.

Capítulo 3. Validación de la propuesta.

- Mínimo: mínimo tiempo en milisegundos de respuesta de una petición.
- Máximo: máximo tiempo en milisegundos de respuesta de una petición.

Prueba #1 Informe agregado

Tabla involucrada: archivo_multimedia.

Consulta: SELECT * FROM public."archivo_multimedia";

Usuarios concurrentes: 100

	Media(ms)	Mínimo(ms)	Máximo(ms)	Mediana(ms)
Resultado	300	10	1300	150

Tabla 10: Resultado de la prueba

Usuarios concurrentes: 300

	Media(ms)	Mínimo(ms)	Máximo(ms)	Mediana(ms)
Resultado	1800	21	9400	550

Tabla 11: Resultado de la prueba

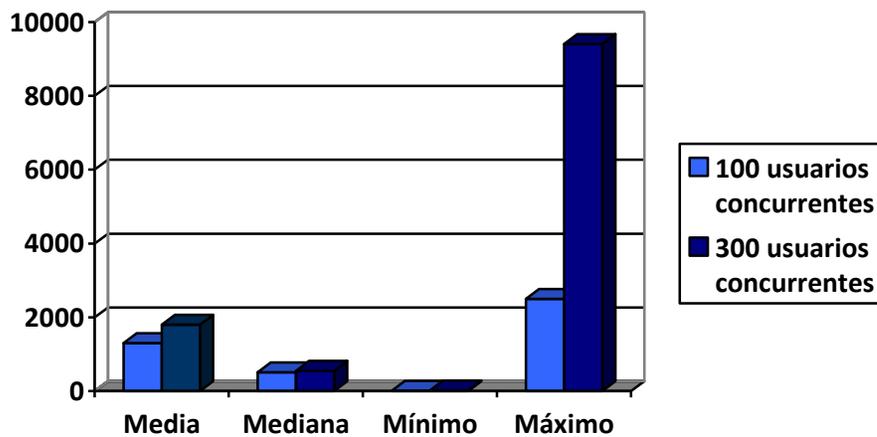


Figura 5 Gráfico del resultado de las pruebas

Capítulo 3. Validación de la propuesta.

Luego de concluidas las pruebas de carga a la base de datos del sistema de gestión y transmisión de contenidos audiovisuales, se obtuvieron resultados satisfactorios. Se obtuvo un tiempo de respuesta óptimo comparado con la gran cantidad de información almacenada para la realización de las pruebas. Los tiempos de respuesta en milisegundos obtenidos después de haber realizado las pruebas de carga a la base de datos muestran un estimado, o sea no deben ser tomados como valores absolutos. Existen numerosos factores que influyen en la rapidez y estabilidad de la base de datos. Un ejemplo es la cantidad de usuarios conectados simultáneamente que mientras este sea mayor, el tiempo de respuesta aumentará. Otro factor es el grado de concurrencia, la mejor forma de probarlo es mediante la interrelación directa con el usuario.

Las pruebas de estrés: se utilizan con el fin de cargar de trabajo a la base de datos con régimen superior al esperado en la explotación real. Se efectúa para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada. Permiten documentar las condiciones bajo las cuales el sistema puede fallar. Verifican la aceptabilidad del desempeño de la base de datos ante condiciones anormales o extremas, dígase volumen de usuarios o transacciones extremadamente alto y escasos recursos. Estas pruebas también se realizaron con la herramienta de Jmeter ingresándole valores superiores para verificar donde era vulnerable la base de datos, concluyendo que el máximo de usuarios concurrentes es 700.

3.12 Trazabilidad de las acciones

La trazabilidad de las acciones tiene la peculiaridad de trazar o dejar huellas de los movimientos y procesos por los que pasa cierto producto. Además, posee una organización o sistema para rastrear, reconstruir o establecer relaciones entre objetos monitoreados, para identificar y analizar situaciones específicas o generales. Un sistema de gestión y transmisión de contenidos audiovisuales mantiene el control sobre todas las acciones que los usuarios realizan sobre la base de datos, es decir si un usuario actualiza un registro, debe quedar constancia de cómo lo ejecutó y el cambio que realizó. De esta forma se logra el seguimiento de las acciones ejecutadas, conformando un historial. Para seguimiento de las trazas en la base de datos se crea la tabla llamada `sf_guard_user`, en la cual se almacena nombre, dirección, correo, fecha de creada, fecha actualizada y datos del usuario, quedando confeccionada la traza en esta tabla.

3.13 Conclusiones parciales

En este capítulo se realizó la validación teórica y funcional de la base de datos para la plataforma de transmisión de contenidos multimedia del centro GEYSED. Se definieron un conjunto de restricciones de integridad, poniendo ejemplos de cómo se ve reflejado esta en la base de datos de un sistema de gestión y transmisión de contenidos audiovisuales a la hora de verificar la consistencia de los datos. Se aplican las tres formas normales acorde a los principios de fraccionamiento de la información en el almacenamiento consistente y perdurable de la misma. Por último a la base de datos se le realizó pruebas con la herramienta Jmeter para los test de carga y estrés, y así verificar que no existan errores de integridad y pruebas de volumen con EMS Data Generator. Queda demostrando que el gestor utilizado y el diseño de las estructuras de la base de datos soportan completamente el almacenamiento de los niveles de información requeridos para su puesta en práctica.

Conclusiones Generales

Después de haber realizado el diseño e implementación de la base de datos para la plataforma de un sistema de gestión y transmisión de contenidos audiovisuales, y comprobar que garantiza la seguridad de los datos, se concluye que se obtuvieron los resultados enumerados a continuación:

- Se seleccionó el modelo, el tipo de base de datos, y el SGBD más adaptable a utilizar en un sistema de gestión y transmisión de contenidos audiovisuales.
- Se obtuvo un diagrama de clases persistentes que cumple con las necesidades de un sistema de gestión y transmisión de contenidos audiovisuales.
- Se normalizó hasta la 3FN y se minimizó la información redundante en la misma.
- Se utilizó una estrategia de replicación que cumple con las características de la arquitectura física de un sistema de gestión y transmisión de contenidos audiovisuales.
- La base de datos respondió satisfactoriamente a las pruebas que se le realizaron.

Con las conclusiones antes expuestas se puede asegurar que la base de datos es eficiente, asegurando la integridad y la consistencia de los datos.

Recomendaciones

Se propone analizar la inclusión en la solución de un balanceador de carga para repartir las tareas entre todos los servidores reales que componen el clúster.

Se propone analizar el rendimiento del sistema con el uso de procedimientos almacenados para valorar la factibilidad de este recurso en aras de contribuir al mejor manejo de los datos entre las capas lógicas y físicas de la aplicación que haga uso de la base de datos.

Bibliografía

AIN. 2011. [En línea] 25 de 10 de 2011. [Citado el: 26 de 11 de 2011.] <http://www.cubarte.cult.cu/periodico/noticias/158795/158795.html>.

Alvarez, Lic. Norberto. *Base de Datos* .

Andrés. 2001. *Arquitectura de Base de Datos*. España : s.n., 2001.

Anzaldo, Juan. 2005. Breve Historia de las Bases de datos. [En línea] 6 de 12 de 2005. [Citado el: 5 de 11 de 2011.]

Arévalo, Mauricio. 2010. [En línea] 2010. <http://es.scribd.com/doc/63764883/33/Ventajas-y-Desventajas-de-MySQL>.

Baixauli, Vicente. 2005. [En línea] 2005. [Citado el: 26 de 11 de 2011.] <http://www.slideshare.net/da4equipo3/historia-de-las-bases-de-datos>.

Barajas, Matilde Moreno. 2009. EL SERVICIO DE CONSULTA DINÁMICA. [En línea] 23 de enero de 2009. [Citado el: 15 de 11 de 2011.] www.gestiopolis.com/administracion-estrategia/servicio-de-consulta-olap-bases-de-datos.htm.

Bello, Claudia E. *Manual de Seguridad en Redes*.

Camallea, Noel L.Nuñez. 2004. Gestión de bases de datos con ado.net. *Gestión de bases de datos con ado.net*. s.l. : Científico técnica, 2004.

Garcia, Ruber hernandez. 2007. *Sistema Automatizado de Teletexto para la Plataforma de Television digital Satelital Cubana*. 2007.

Graells, Dr. Pere Marquès. 2000. LAS TIC Y SUS APORTACIONES A LA SOCIEDAD. *LAS TIC Y SUS APORTACIONES A LA SOCIEDAD*. [En línea] 2000. <http://peremarques.pangea.org/tic.htm>.

Hall, Prentice. 2008. *Intruduccion a los Sistemas de Base de datos*. UNIVERSIDAD FEDERICO SRNTA MARIA : s.n., 2008.

Henry F. Korth, Abraham Silberschatz. *FUNDAMENTOS DE BASES DE DATOS*.

Iborra, Marcelo J Armengot. 2004. *Sistema de Base de Datos cliente-servidor con interfaz gráfica multiplataforma.* Valencia : s.n., 2004.

M, Pablo Y. Portal de Seguridad Informática. [En línea] <https://seguridad.uci.cu/>.

Magazine, Inc. 2009. 2009, Revista Ciencia e Ingeniería, págs. Vol. 30, No 2, pp. 149-160.

Mora, Roberto Canales. 2004. [En línea] 02 de 02 de 2004. [Citado el: 25 de 11 de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.

Neustadt Wiener. 2008. CyberCluster. [En línea] 2008. [Citado el: 28 de abril de 2012.] http://www.cybertec.at/download/dokumentation/documentation_cybercluster.pdf.

Ostorga, Victor. 2010. *Replicación de Bases de datos PostgreSQL.* 2010.

Paré, Rafael Camps. *Introducción a las Bases de Datos.* Barcelona : s.n.

pgadmin. 2010. pgadmin PostgreSQL Tools. *pgadmin PostgreSQL Tools.* [En línea] 2010. [Citado el: 28 de 11 de 2011.] <http://www.pgadmin.org/>.

PostgreSQL, EMS Data Generator for. 2008. EMS Data Generator for PostgreSQL. [En línea] 7 de 10 de 2008. [Citado el: 4 de 4 de 2012.] <http://www.sqlmanager.net/products/postgresql/datagenerator>.

Pressman, Rogers. 2010. *Software Engineering a Practitioner's Approach.* New York San Francisco : s.n., 2010. 7 ma versión.

rafaelma. 2010. PostreSQL_es. [En línea] 29 de 06 de 2010. www.postgresql.org.

Rocks. 2011. Open_source toolkit for real and virtual clusters. [En línea] abril de 2011. <http://www.rocksclusters.org/wordpress/>.

Rosa, Karen Mercedes Zerquera. 2009. *PRIMICIA, Plataforma de Televisión Informativa. Rol de Diseñador de Bases de Datos.* La Habana : s.n., 2009.

Rozic, Sergio Ezequiel. 2004. *Bases de datos y sus aplicaciones con SQL.* Buenos Aires, Argentina : s.n., 2004.

Sabater, Jaume. 2008. Replicación y alta disponibilidad de PostgreSQL con pgpool-II. [En línea] 2008. <http://linuxsilo.net/articulos/postgresql-pgpool.html>..

Savedra, Juan. 2009. *Integridad Informática enfocada a las bases de datos.* 2009.

Server C.d.D.d.S. 2009. Integridad de los datos. [En línea] 2009. [Citado el: 26 de 3 de 2012.] <http://msdn.microsoft.com/es-es/library/ms184276.aspx>..

Vega, Norge Fajardo. 2007. *Sistema de réplica para bases de datos distribuidas en PostgreSQL.* La Habana : s.n., 2007.

Velthuis, Mario Piattini. 2000. *Líneas de evolución de las.* 2000.

Victoria, msc. definicionabc. [En línea] [Citado el: 28 de abril de 2011.] <http://www.definicionabc.com/tecnologia/cluster.php>.

w.Hansen, Gary. *Diseño y administración de las bases de datos.*

Wieggers, Karl E. 2003. *Software Requirements.* 2003.

Wordpress. 2008. Definición.de. *Definición.de.* [En línea] 2008. <http://definicion.de/integridad/>.

Referencias Bibliográficas

AIN. 2011. [En línea] 25 de 10 de 2011. [Citado el: 26 de 11 de 2011.] <http://www.cubarte.cult.cu/periodico/noticias/158795/158795.html>.

Alvarez, Lic. Norberto. *Base de Datos* .

Andrés. 2001. *Arquitectura de Base de Datos*. España : s.n., 2001.

Anzaldo, Juan. 2005. Breve Historia de las Bases de datos. [En línea] 6 de 12 de 2005. [Citado el: 5 de 11 de 2011.]

Barajas, Matilde Moreno. 2009. EL SERVICIO DE CONSULTA DINÁMICA. [En línea] 23 de enero de 2009. [Citado el: 15 de 11 de 2011.] www.gestiopolis.com/administracion-estrategia/servicio-de-consulta-olap-bases-de-datos.htm.

Bello, Claudia E. *Manual de Seguridad en Redes*.

Camallea, Noel L.Nuñez. 2004. Gestión de bases de datos con ado.net. *Gestión de bases de datos con ado.net*. s.l. : Científico técnica, 2004.

Garcia, Ruber hernandez. 2007. *Sistema Automatizado de Teletexto para la Plataforma de Television digital Satelital Cubana*. 2007.

Graells, Dr. Pere Marquès. 2000. LAS TIC Y SUS APORTACIONES A LA SOCIEDAD. *LAS TIC Y SUS APORTACIONES A LA SOCIEDAD*. [En línea] 2000. <http://peremarques.pangea.org/tic.htm>.

Hall, Prentice. 2008. *Intruduccion a los Sistemas de Base de datos*. UNIVERSIDAD FEDERICO SRNTA MARIA : s.n., 2008.

Henry F. Korth, Abraham Silberschatz. *FUNDAMENTOS DE BASES DE DATOS*.

Iborra, Marcelo J Armengot. 2004. *Sistema de Base de Datos cliente-servidor con interfaz gráfica multiplataforma*. Valencia : s.n., 2004.

M, Pablo Y. Portal de Seguridad Informática. [En línea] <https://seguridad.uci.cu/>.

- Magazine, Inc. 2009.** 2009, Revista Ciencia e Ingeniería, págs. Vol. 30, No 2, pp. 149-160.
- Mauricio Arévalo M. 2010.** [En línea] 2010. <http://es.scribd.com/doc/63764883/33/Ventajas-y-Desventajas-de-MySQL>.
- Mora, Roberto Canales. 2004.** [En línea] 02 de 02 de 2004. [Citado el: 25 de 11 de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.
- Neustadt Wiener. 2008.** CyberCluster. [En línea] 2008. [Citado el: 28 de abril de 2012.] http://www.cybertec.at/download/dokumentation/documentation_cybercluster.pdf.
- Ostorga, Victor. 2010.** *Replicación de Bases de datos PostgreSQL*. 2010.
- Paré, Rafael Camps.** *Introducción a las Bases de Datos*. Barcelona : s.n.
- pgadmin. 2010.** pgadmin PostgreSQL Tools. *pgadmin PostgreSQL Tools*. [En línea] 2010. [Citado el: 28 de 11 de 2011.] <http://www.pgadmin.org/>.
- PostgreSQL, EMS Data Generator for. 2008.** EMS Data Generator for PostgreSQL. [En línea] 7 de 10 de 2008. [Citado el: 4 de 4 de 2012.] <http://www.sqlmanager.net/products/postgresql/datagenerator>.
- rafaelma. 2010.** PostreSQL_es. [En línea] 29 de 06 de 2010. www.postgresql.org.
- Rocks. 2011.** Open_source toolkit for real and virtual clusters. [En línea] abril de 2011. <http://www.rocksclusters.org/wordpress/>.
- Rosa, Karen Mercedes Zerquera. 2009.** *PRIMICIA, Plataforma de Televisión Informativa. Rol de Diseñador de Bases de Datos*. La Habana : s.n., 2009.
- Rozic, Sergio Ezequiel. 2004.** *Bases de datos y sus aplicaciones con SQL*. Buenos Aires, Argentina : s.n., 2004.
- Sabater, Jaume. 2008.** Replicación y alta disponibilidad de PostgreSQL con pgpool-II. [En línea] 2008. <http://linuxsilo.net/articles/postgresql-pgpool.html>.
- Savedra, Juan. 2009.** *Integridad Informática enfocada a las bases de datos*. 2009.

Server C.d.D.d.S. 2009. Integridad de los datos. [En línea] 2009. [Citado el: 26 de 3 de 2012.] <http://msdn.microsoft.com/es-es/library/ms184276.aspx..>

Vega, Norge Fajardo. 2007. *Sistema de réplica para bases de datos distribuidas en PostgreSQL.* La Habana : s.n., 2007.

Velthuis, Mario Piattini. 2000. *Líneas de evolución de las.* 2000.

Vicente Baixauli. 2005. [En línea] 2005. [Citado el: 26 de 11 de 2011.] <http://www.slideshare.net/da4equipo3/historia-de-las-bases-de-datos>.

Victoria, msc. definicionabc. [En línea] [Citado el: 28 de abril de 2011.] <http://www.definicionabc.com/tecnologia/cluster.php>.

w.Hansen, Gary. *Diseño y administración de las bases de datos.*

Wieggers, Karl E. 2003. *Software Requirements.* 2003.

Wordpress. 2008. Definición.de. *Definición.de.* [En línea] 2008. <http://definicion.de/integridad/>.

Glosario de términos

Licencia GPL: es español significa licencia publica general, está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su intención es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que limiten esas libertades a los usuarios.

La licencia BSD: (Berkeley Software Distribution) es una familia de licencias de software libre permisiva, permite el uso del código fuente en software no libre.

Registro: está formado por todos los campos referidos a una entidad u objeto almacenado.

Campos: cada uno de los elementos que componen un registro.

Software: es un elemento fundamental dentro de las tecnologías de la información, ya que esta es la en parte intangible de sus productos y en parte por estar sometida a un rápido y continuo proceso de cambio tecnológico. Según la ISO (International Standardisation Organisation) y la WIPO (World Intellectual Property Organisation), dice que un software se entiende por la “producción de un conjunto estructurado de instrucciones, procedimientos, programas, reglas y documentación contenida en distintos tipos de soporte físico (cinta, discos, circuitos eléctricos), con el objetivo de hacer posible el uso de equipos de procesamiento electrónico de datos”.

Aplicaciones: se pueden considerar como un programa informático que le sirve al usuario como herramienta para realizar un trabajo o varios a la vez, diferente a los programas como los sistemas operativos que estos se encargan de que el ordenador funcione. Existiendo varios tipos de aplicaciones como las web y las de escritorio.

Transacciones: son un programa que se ejecuta como una sola operación.

Herramientas CASE: (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

URL: (Uniform Resource Locator, Localizador Uniforme de Recurso): Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos como documentos e imágenes en Internet para su localización.

UNIX: Es un sistema operativo de tiempo compartido, controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios correr sus programas.

TCP: (Transmission Control Protocol, Protocolo de Control de Transmisión). Protocolo de comunicación orientado a conexión y fiable ubicado en el nivel de transporte del modelo OSI, es uno de los protocolos fundamentales en Internet.

JDBC: (Java Database Connectivity) es un API para trabajar con bases de datos desde Java, independientemente de la base de datos a la que se acceda.

IP: (Internet Protocol, Protocolo de Internet). Protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Atributo compuesto: pueden ser divididos en pequeñas partes, las cuales representan atributos básicos con existencia independiente.

Atributo multivaluado: un atributo es multivaluado cuando para una misma entidad puede tomar varios valores diferentes, con independencia de los valores que puedan tomar el resto de los atributos.

Anexos

Anexo #1: descripción de tablas de la base de datos del sistema de gestión y transmisión de contenidos audiovisuales.

Anexo #2: manual de réplica con la técnica Hot Standby-Streaming Replication.